

VEGOVO



STROKOVNA REVIJA
2016/17

06

RAČUNALNIŠTVO

44

ELEKTROTEHNIKA

60

MEHATRONIKA

64

NARAVOSLOVJE

78

MATEMATIKA

86

IN ŠE TO

Šolsko glasilo Vegovc 2017
Izdala: Vegova Ljubljana, Vegova 4

Uredniki:

Nejc Bertoncelj, G 2. A
Jernej Jezeršek, G 2. A
Žan Oberstar, G 2. A
Domen Klinc, G 2. B
Tine Venko, G 2. B
Matej Pevec, G 2. B
Matej Poljanšek, G 2. B

Tiskano: COLLEGIUM GRAPHICUM, Ljubljana

Mentorici:

Alenka Zorko, prof.
Mojca Stare Aljaž, prof.

Naklada: 300 izvodov



Dijaki in profesorji!

Naj vas debelina tokratnega Vegovca 2017 ne odvrne od branja, saj smo si letos v uredništvu zastavili drugačen cilj in pripravili za vas strokovno revijo. Zanja so svoje članke prispevali tako dijaki strokovne gimnazije in strokovne šole kot tudi učitelji in drugi delavci šole.

Ja, prav ste prebrali; letošnji Vegovc je v glavnem strokovna publikacija, kar pomeni, da so v njem objavljeni članki s strokovnih področij, ki so v ospredju na naši šoli. Seveda smo vključili tudi poljudna in umetnostna besedila in kakšen lahkotnejši članek. Ops, tiskarski škrat je spet udaril! Mislili smo seveda članek. Saj veste, kako je to, ko se kar naenkrat pojavi in nam jo zagode, potem smo pa mi krivi za njegove napake.

Upamo torej, da nam tokrat ni preveč ponagajal, in želimo, da bo branje našega in vašega Vegovca v užitek. Pa naj bo tudi poučno.

Tiskarski škrat

Včasih se je dan s časopisom začel s tiskarskim škratom. Vsaj na eni strani je nosil krivdo za napako. Na vojno ozemlje so vdrl tuji piščanci namesto plačanci, iz svetovnega kaosa je nastal svetovni kavs in na straneh kulture so predstavili knjigo V objemu s prascem namesto s prsrcom, kot je bil njen naslov, pa karkoli naj bi to že pomenilo. Škratka, tiskarski škrati so bili ves čas med nami in prav lahko si jih je bilo predstavljati, kako ubogim tiskarjem, ki s pincetko zlagajo kovinske črke, skačejo pod prste. S stavkom »tiskarski škrat nam jo je (spet) zagodel«, je bilo naslednji dan vse oproščeno.

Tisti, ki smo rastle še v pravih časopisnih časih, smo rastle skupaj s škrti in nekateri med njimi so postali prave legende. Morda jih je čas nabuhnil, morda so resnični. Med večje zgodbe sodijo škrti iz sedemdesetih, osemdesetih let trdega režima, ko je dan borca postal dan norca, Edvard Kardelj pa Kradelj. Škrat iz nekrologa za istim politikom socializma je pokojniku nadel poseben »posluh za ljudi, sodelavce in sobarice«.

Mišljen je bil seveda posluh za soborce. Na naslovnici nedeljskega časopisa so poslanci podrli Bajuka namesto podpri, češke psice pa so bili v resnici češki pisci na pisateljskem srečanju. Starejši kolegi se spomnijo, kako so v tiskarni nekoč ob predolgem članku tiskarji kar sami odrezali kos besedila in se je potem končal s stavkom: Za vse težave koroških Slovencev je kriv ... sledilo je ime podpisanega novinarja, reveža, ki so ga morali v nočnih urah krajšati. Novinarka, ki je redno pokrivala novice s tržnice, je nekoč kunčje meso natipkala kot kurčje meso.

Danes namesto škratov vse skupaj opravimo s popravkom in opravičilom, saj vendar vemo, da za računalnikom ne sedi bradato bitje s kapico. Nič bolj romantičnega si ne znamo izmisliti v teh digitalnih časih, ko se nam kljub vsem mogočim črkovalnikom še vedno lahko primeri naslov »Potiti se je človeško«. Da, motiti se je človeško. Ali kot je kolega hudomušno zapisal svoj popravek: »Ponagajal nam je tiskarski škrat. Prijeli ga bomo za ušesa.« **TANJA JAKLIČ**

Delo, 8. 10. 2016



Posnetek temne meglice Konjska glava v ozvezdju Orion. Na posnetku vidimo tudi sled umetnega satelita. Skupni čas osvetlitve 7 min. Foto: R. Snoj

Kaj vse se zgodi v eni sekundi na internetu

Tim Lunar, R 3. A

Vsako sekundo dneva je tako ali drugače omreženih v splet kar 46,1 odstotka prebivalstva Zemlje oziroma 3,4 milijarde ljudi.

V kratkem času sekunde Google-va mašinerija opravi 54.000 iskanj na zahtevo uporabnikov.

Objavi se okoli 7000 sporočil na Twitterju in pošlje več kot dva milijona sporočil prek e-pošte.

Vsako sekundo je na Facebooku 20.000 ljudi, pet novih zemljanov pa ravno odpira nov račun.

Na Instagram je naloženih 729 novih fotografij, na Youtubu se odvrti 125.406 videov, po Skypu pa se opravi 2177 klicev.

V sekundi se po spletu pretoči za 36.854 gigabajtov podatkov. Za tisto, kar je na začetku potreboval ves dan, danes potrebuje dve desetinki.

Kako silovito rast je doživel splet, dokazuje primer Googla. Če danes vsako sekundo odgovori na 54.000 iskanj, kar je 90 milijard na dan, je še leta 2006 v eni sekundi opravil "le" 10.000 iskanj.

Trenutno je na svetu že 1,05 milijarde različnih spletnih strani, samo danes so ugrabili že 56.000 strani.

Zaradi interneta je človeštvo porabilo prek 2,7 milijona megavatnih ur električne energije. Stranski produkt je tudi 2,5 milijona ton CO₂.

V kako drugačno dobo je človeštvo stopilo z internetom, priča še en podatek, ki je bil objavljen leta 2013 in je torej veljal še za takratni čas: več kot 90 odstotkov vseh podatkov, ki jih je ustvarilo človeštvo od najzgodnejših začetkov pisave do takratnih dni, je bilo ustvarjenih v zgolj zadnjih dveh letih.

Posamezni uporabnik svetovnega spleta si težko predstavlja razsežnosti elektronskih komunikacij, ki danes mrežijo svet. Vendar je dejstvo, da kakih dvajset let po začetku silovitega vzpona interneta danes čisto vsak trenutek na milijone ljudi tako ali drugače uporablja svetovni splet bodisi za brskanje, pošiljanje e-pošte oziroma sporočil prek socialnih omrežij bodisi za gledanje vsebin prek pretočnih omrežij ...

<http://www.racunalniske-novice.com/novice/dogodki-in-obvestila/poglejte-kaj-vse-se-zgodi-na-internetu-v-picli-sekundi.html>

<https://www.domovina.je/kaj-vse-se-na-internetu-zgodi-v-eni-sami-sekundi/>

<http://www.zurnal24.si/vse-to-se-v-zgolj-sekundi-zgodi-na-internetu-clanek-275243>

Generacije omrežij

Jure Časar, R 3. A

1G – prva generacija

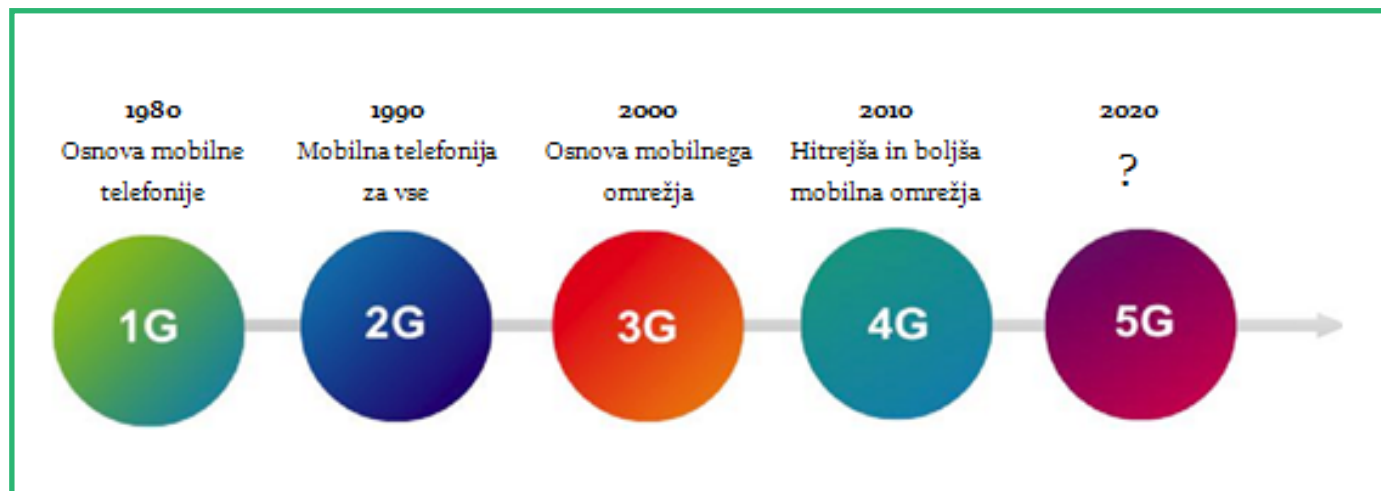
Prva generacija mobilnega omrežja je podpirala analogne storitve za mobilne telefone. Te storitve so bile samo za zvočno komuniciranje, tehnologija ni podpirala SMS ali drugih podatkovnih storitev. 1G je deloval na principu preklopnega vezja. To pomeni, da je bila povezava pri klicu vzpostavljena in obdržana, dokler je ni klicatelj prekinil. Prvi mobilni telefoni 1G

so se pojavili leta 1980 in so imeli zelo kratko dobo baterije, prav tako niso poznali kasnejših storitev druge generacije 2G, kot so SMS-sporočila ali kakšne druge podatkovne storitve.

2G – druga generacija (GSM, GPRS, EDGE)

Leta 1991 se je pojavilo prvo 2G-omrežje, in sicer na Finskem, ki je omogočalo prenos podatkov

v digitalni obliki in s tem tudi postavilo nekatere temelje za današnje brezžične sisteme. Kasnejši razvoj je že omogočil pošiljanje SMS in večpredstavnostnih sporočil kakor tudi brskanje po spletu. Slednje je na 2G-telefonu zelo počasno, hitrosti je mogoče primerjati s klicnim dostopom. Kadar smo z 2G-telefonom povezani v splet, ne moremo klicati ali sprejemati klicev.



3G – tretja generacija (UMTS, HSPA+)

Japonska je leta 2001 prva začela uporabljati omrežje tretje generacije 3G. Sledila ji je Evropa leta 2003 in ZDA kmalu zatem. To omrežje zagotavlja storitve, ki jih uporabljamo danes. Brskamo po spletu in beremo elektronsko pošto pri zadovoljivi hitrosti. Omrežje 3G večinoma omogoča klicanje in brskanje po spletu. Telefon Apple iPhone je močno vplival na način brskanja po spletu. Prav tako so se na veliko začela uporabljati brezžična omrežja. Zaradi vse večjih pričakovanj in potreb po brezžični mobilnosti se je že leta 2008 začel razvijati standard 4G, ki obljublja še veliko višje hitrosti prenosa podatkov.

LTE (4G) – četrta generacija

Čeprav LTE (Long-Term Evolution) pogosto označujejo kot četrto generacijo mobilnih omrežij (4G), ne ustreza originalni definiciji 4G, kot jo je sprva zasnovala organizacija ITU (International Telecommunication Union), zato je LTE bolj primerno opredeliti kot predhodnika prave 4G. ITU

je kljub temu s posebno izjavo dovolil oznako 4G tudi za LTE in nekatere druge predhodnike prave 4G. Ta bo na voljo šele s standardom LTE Advanced.

Poglavitna prednost LTE je v doseganju večjih hitrosti, ki naj bi teoretično dosegale 326,4 Mbit/s v smeri proti uporabniku in 86,4 Mbit/s v nasprotni smeri. Gre za izključno paketno omrežje, saj v standardu ni definiran prenos govora. Prvo komercialno dostopno omrežje LTE je postavil operater TeliaSonera v Stockholmu in Oslu decembra 2009. LTE lahko deluje na različnih frekvencah, med drugim na 700 (ZDA), 800, 1800 in 2600 MHz.

V Sloveniji je bilo prvo LTE-omrežje na voljo 2013.

Sedaj se pojavljajo že špekulacije o "prihodu" omrežja 5G oz. nekatera podjetja so že uspela vzpostaviti in testirati povezavo 5G (Samsung). "Samo" 70 % ljudi je namreč zadovoljnih s povezavo 4G, kajti nekateri pravijo, da je prepočasna.

Mnogi menijo, da naj bi bilo omrežje 5G na voljo po letu 2020.

Veliki telefonski ponudniki v ZDA in po svetu si po predstavitvi omrežja 5G, ki pa še ni dovolj optimizirano za javnost, želijo let tega čim prej predstaviti v svojih ponudbah. Toda da bi dokončno osvojili 5G-omrežje, morajo ponudniki in izdelovalci pametnih telefonov ugotoviti, kako "dostaviti" podatke z visoko hitrostjo uporabnikom, ki se vozijo z avtomobili, vlaki ali pa celo hodijo po pločnikih. Toda 5G-omrežje ni namenjeno samo telefonskim uporabnikom, ampak tudi npr. avtonomnim avtomobilom, industrijskim robotom ipd.

Nekateri ameriški ponudniki pravijo, da bi že letos lahko izdali 5G-omrežje, vendar ne tako dovršeno, kot bi moralo biti.

<http://www.hamsterdesign.si/blog/generacije-mobilnih-omrezij-1g-2g-3g>

<http://www.delo.si/gospodarstvo/infrastruktura/5-g-pilotno-v-mobilne-komunikacije.html>

<http://spectrum.ieee.org/telecom/wireless/verizon-and-att-prepare-to-bring-5g-to-market>

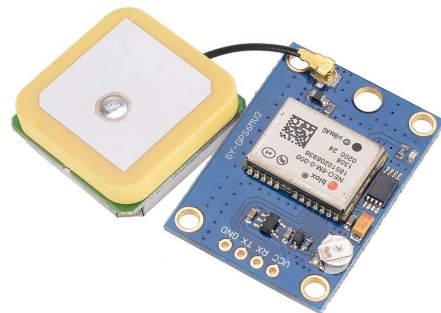
<https://krify.co/tag/migrate-4g-to-5g-the-next-big-network/>

Fantastični baloni in kako jih najti

Jan Leskovec, G 2. B

Junija 2017 bomo G 2. B spustili meteorološki balon z nekaj senzorji, ki bodo zapisovali podatke o ozračju. Balon, napolnjen s helijem, bo nenadzorovano potoval po zraku in po približno dveh urah pristal. Prvo vprašanje, ki se porodi ob pomisleku, da bo balon poletel 30 km visoko in daleč izven dosega pogleda, je, kako bomo ta balon po pristanku našli. To težavo bomo reševali s sledilnim sistemom, ki je sestavljen iz GPS-modula, mikroprocesorja Raspberry Pi, brezžičnega modema in mobilne aplikacije za sistem Android. Sistem bo deloval tako,

da bo Raspberry Pi bral podatke o lokaciji z GPS-modula Ublox NEO-6M in te podatke pošiljal preko brezžičnega modema na strežnik, iz katerega pa bo naša aplikacija SkyTracker prebrala trenutno lokacijo balona in to lokacijo narisala na zemljevidu Google Maps. Kasneje lahko aplikaciji tudi dodamo algoritme, ki bodo na podlagi časa in lokacije izračunali trenutno hitrost balona in predvidevanje pristajalnega mesta. Seveda zaradi omejitev moči signala mobilnega omrežja ne bo mogoče sprejemati lokacije balona, ko bo izven dosega oddajnikov, a ko se



bo balon začel spuščati in ga bomo začeli iskati, bo ponovno prišel v domet oddajnikov mobilnega omrežja; tako bomo ponovno videli lokacijo našega balona, da ga bomo lahko našli in iz njega vzeli podatke, ki jih je zapisal na potovanju skozi ozračje.

Internet stvari (IoT) – prihodnost ali utvara?

Manca Jurančič, R 3. A

Leta 1999 je britanski inženir Kevin Ashton prvič uporabil besedno zvezo internet stvari (IoT). Odkar človeštvo uporablja medmrežje, nam internet stvari prinaša največ sprememb. Pravijo, da gre za pravo revolucijo svetovnega spleta.

Preko interneta komuniciramo predvsem ljudje, medtem ko je komunikacija med napravami in stroji še v povojih. Kar 99 % električnih naprav še ni povezanih v internet, tiste, ki pa so, so zelo omejene (kot na majhne otočke). Predvidevajo, da bo do leta 2020 v splet in med seboj povezanih 30 milijard naprav (50 %). Po zaslugi milijard senzorjev, ki so vgrajeni v naprave, se bo oblikovala nova dimenzija inteligence v praktično vseh gospodarskih panogah. Bolje naj bi razumeli naravo, zdravila, medicino, avto-

mobili nas bodo vozili sami, ... Vsa ta inteligenca naj bi bila shranjena v podatkovnih zbirkah in je uporabna za boljšo kakovost življenja. Trenutno se največ IoT-ja uporablja v transportu in logistiki, najdemo ga tudi že v energetiki. Danes v internet stvari povezujemo hladilnike, rastline, avtomobile, proizvodne stroje, okna, obleke oz. praktično vse, kar uporabljamo v vsakdanjem življenju. Vse več stvarjem dodajamo senzor ali kompleksni mikroračunalnik, ki prek brezžične povezave komunicira z okoljem, s pridobljenimi podatki in njihovo obdelavo pa ustvarjajo nadaljnje akcije, ki komunicirajo z drugimi stvarmi (M2M oz. »machine to machine«) ali s človekom. Takih aplikacij, ki interaktivno komunicirajo z napravami in stvarmi okoli nas, je že danes ogromno in jih je mogoče najti

tudi v bližnji okolici. Primer takih enostavnih, a uporabnih aplikacij je na primer prikaz razpoložljivosti oz. zasedenosti parkirišč v mestu, aktualni prikaz prihodov mestnih avtobusov na posamezna postajališča ali pa spremljanje aktivnosti posameznika skozi celoten dan in prikaz uporabe energije (kalorij). Toda postavlja se tudi vprašanje, ali je uporaba IoT nova revolucija ali pot v kaos. Obenem vse več strokovnjakov opozarja, da utegnejo biti te naprave pravcati recept za kaos. Nekateri celo menijo, da bo zamisel povsem pogorela, če se izdelovalci tega segmenta ne bodo lotili preudarneje.

V današnjih časih vse več izdelkov okoli nas dobiva pridevnik »pametni«. Seveda s tem želijo predvsem prepričati kupce, da bi z navduše-

njem zamenjali svoje še povsem dobre izdelke. Analitske družbe naprave IoT kujejo v zvezde. Te bodo namreč preproste, z majhno dodatno vrednostjo za izdelovalca, potem ko jih bo prodal in oddal kupcem. Zelo verjetno bo izdelovalec kaj kmalu pozabil nanje in jih nehal posodabljati in vzdrževati vgrajeno sistemsko programsko opremo.

Poleg tega imajo naprave IoT nadvse primitivno zaščito. V številne je mogoče vdreti s privzetim geslom ali z najosnovnejšimi tehnikami vdora.

Varnost pa ni edina težava naprav IoT. Številni strokovnjaki opozarjajo na skoraj popolno pomanjkanje standardov za medsebojno konfiguracijo in povezovanje teh naprav. Jean-Louis Gasee (eden ključnih direktorjev iz zlate dobe družbe Apple) je v svoj blog zapisal, da se bo pri napravah IoT pokazal sindrom »košarice daljincev«. Skoraj vsi imamo doma cel kup daljincev. Toliko jih imamo, ker se izdelovalci niso mogli ali hoteli domeniti o enotnem standardu. Velika verje-



tnost je, da bo tako tudi pri napravah IoT. Morda enotnih standardov ne bodo nikoli dosegli. Brez njih pa se nam obeta kaos. Če bodo naprave IoT vsaj približno uspešne, kot napovedujejo, bo zanj zmanjkalo internetnih naslovov. Z drugimi besedami, uporabljati bomo morali naslov IPv6.

Zamisel internetnih stvari je sama po sebi povsem dobra, spodbuja nove načine razmišljanja in uporabe tehnologije. Toda zdi se, da snovalci za zdaj puščajo preveč področij ne-

dorečenih, da bi bila ta tehnologija uspešnica na daljši rok. To pa ne pomeni, da se stvari ne morejo obrniti na bolje. Dobro bi bilo, da bi industrijski velikani na ta odprta vprašanja verodostojno odgovorili zdaj, še preden trg preplavijo z nedorečenimi in nedokončanimi izdelki.

<http://www.monitor.si/clanek/iot-revolucija-ali-pot-v-kaos/154188/>
<https://goo.gl/8y0KEE>
<https://s-media-cache-ak0.pinimg.com/originals/4d/34/>

Računalniški prstni odtis

Nejc Klopčič, R 3. A

Vsem je jasno, da so piškotki namenjeni prepoznavanju obiskovalcev na spletnih straneh, kar ima v večini primerov povsem legitimne razloge in uporabne namene. Toda četudi pobrišemo piškotke in računalnik priključimo v drugo omrežje, s čimer dobimo nov naslov IP, nas spletne strani še vedno z veliko verjetnostjo lahko identificirajo. Raziskovalci z Univerze Lehigh in Washington v St. Louisu so pokazali, kako lahko z 99,24-odstotno zanesljivostjo

enolično identificirajo uporabnike, četudi ne poznajo njihovega IP-naslova in imajo izklopljene piškotke in skripte. Tudi če so uporabljali različne brskalnike na isti napravi, so še vedno v 83 odstotkih primerov ugotovili, da gre za isti računalnik.

Kako dobro lahko prepoznajo vaš računalnik, lahko preverite na spletni strani Unique Machine, ki so jo postavili kot demonstracijo delovanja razvite metode. Upo-

rabljajo predvsem identifikacijo prek WebGL, in sicer analizirajo podrobnosti pri izrisu tekstur, antialiasing in druge grafične podrobnosti, ki so v kombinaciji zelo specifične za posamezen računalnik. Pri tem ni



pomembno, ali se izračunavanje dogaja dejansko na grafični kartici ali pa gre za programsko emulacijo, saj je v obeh primerih rezultat unikaten. Nekaterе druge operacije, denimo manipulacije z JPG in PNG, pa so odvisne od brskalnika, zato spletna stran izpiše dva prstna odtisa - za brskalnik, ki se ne spreminja niti z brisanjem piškotkov, in za ra-

čunalnik, ki ostane enak tudi ob uporabi drugega brskalnika.

Zastavi se vprašanje, ali se lahko tovrstnemu prepoznavanju izognemo. Najbliže temu je brskalnik Tor, ki dostopa do večine informacij o sistemu ne dovoli in nima vključenega Canvasa in WebGL; nekaj podatkov vseeno razkriva, denimo ločljivost za-

slona. Koristno je tudi poganjati brskalnike v navideznih strojih. Ob tem velja omeniti, da omenjeni prispevek ni prvi na tem področju, saj se raziskovalci z identifikacijo na internetu ukvarjajo že dolgo. So pa sedaj prvič celoti izrabili WebGL.

Vir: Matej Huš, www.slo-tech.com/ novice (dostop: januar 2017)

Internet ali ne internet

2. mesto na natečaju Instituta Jožef Stefan v Ljubljani

Matej Poljanšek, G 2. B

(Mentorica: Mojca Stare Aljaž, prof.)

Internet ali ne internet, to je sedaj vprašanje. Vsak uporabnik interneta bi si ga moral zastaviti, saj ima internet, kot vsaka stvar, dobre in slabe strani, ki pa so predvsem odvisne od uporabe. Po eni strani je zelo priročno orodje oziroma pripomoček, ki ga vsi z veseljem uporabljamo, po drugi, slabši strani pa je dom kiberkriminalcev, hekerjev, pedofilov in drugih podobnih oseb z druge strani zakona, zato bi se lahko upravičeno vprašali, ali ga sploh potrebujemo.

Internet, poznan tudi kot medmrežje, je bil ustvarjen z namenom hitrega širjenja informacij med tremi ameriškimi univerzami, kaj hitro pa ga je začela uporabljati tudi ameriška vojska. Prvi, ki so imeli priložnost preizkusiti internet v lastnih domovih, so bili Američani, kaj kmalu pa je postal dostopen tudi dobršnemu delu ostalega svetovnega prebivalstva, približno 2,4 milijarde ljudem. Nadomestil je marsikateri obisk knjižnice, ogled koncerta in podobne dejavnosti. Vse to se je

preneslo v virtualni svet. Nastalo je veliko število spletnih strani, ki ponujajo to in še marsikaj drugega. Vprašati pa se moramo, ali vse to res potrebujemo in kaj nam vse to sploh pomeni. Ali je dobro, da s prijateljem poklepeta preko Skypa, namesto da bi se srečali z njim osebno in s tem vzpostavili osebni odnos, primeren človeka? Ali je res bolje gledati filme na YouTube, namesto da bi šli v kino in s tem spoznali kakšno novo osebo, mogoče celo ljubezen? So res internetne igre zabavnejše od tistih, ki so jih igrali naši dedki in babice, ko so bili še otroci? Mislim, da z vsemi temi medmrežnimi povezovanji že pretiravamo.

Poleg tega, da nas internet odpelje iz resničnega sveta, pa skriva še mnoge druge pasti. Najočitnejša je zasvojenost z internetom, ki jo vsaka država definira po svoje. V Evropski uniji se zasvojenost na primer definira z reakcijami osebe na zunanje dražljaje, kot so izklop računalnika, izpad električnega toka in podobne nevšečnosti za to

osebo, medtem ko se odvisnost na Kitajskem definira le na podlagi časa, preživetega pred računalnikom, in ne na podlagi reakcijskih izpadov. V resničnem svetu pa poleg tega obstajajo še nevarnosti spletnih klepetalnic, kjer se največkrat skrivajo pedofili in drugi podobni izmečki. Skrivajo se za lažnimi profili ali profili znancev žrtve in se z izdajanjem za prijatelje dogovorijo za skrivno srečanje, kjer napadejo žrtev. Obstajajo tudi možnosti hekerskih vdorov, ki najbolj grozijo uglednim posameznikom in podjetjem, ki nato izgubijo svojo zasebnost in pogosto tudi zaupne podatke in celo lastno identiteto; obstaja pa tudi nevarnost, da prestrežejo zaupne dokumente ali podatke.

Za zaščito pred vsemi temi pastmi in nevšečnostmi in za popravilo nastale škode so nastale različne metode, pripomočki in programi. Pri odvisnosti osebo navadno bolj ali manj uspešno zdravimo v različnih ustanovah in kampih, kjer se, podobno kot pri drugih obli-

kah zasvojenosti, na težak način odvadimo oziroma naučimo kontrolirati svoje početje, še bolje pa je, da se vsak posameznik nauči obvladovati potrebo po računalniku, še preden sploh pride do odvisnosti. Za zaščito pred pedofili in posiljevalci je v največji meri odgovoren uporabnik sam, tako da previdno izbira prijatelje in ne izdaja ničesar zaupnega tujcem, še posebno pa ne svojih osebnih podatkov. Če pa že pride do zlorabe, je naloga pristojnih organov, da pomagajo žrtvi predvsem s psihološko pomočjo. Hekerski vdori so malo bolj težavni, saj jih je skoraj nemogoče predvideti, če se zgodi jo izven skupine, ki ji po navadi pripadajo tarče napadov. Za preprečitev vdorov lahko uporabnik

namesti specializirane programe, ki otežijo in celo onemogočijo vdore v sistem. K zaščiti pripomoreta tudi zapleteno geslo in šifriranje, ki oteži vdor in prepreči sporočilo, če to vsebuje podatke, ki jih želimo zadržati med nami in prejemnikom. Najbolj učinkovito je asimetrično šifriranje, ki vsebuje asimetrična ključa, ki preprečita vpogled v podatke celo pošiljatelju, saj se ti lahko dekodirajo le s pomočjo prejemnikovega ključa, ki pa je različen od pošiljateljevega. S tem se zagotovi varnost podatkov.

Vse te pasti bi marsikoga prestrašile. Drži, da je internet nevarno igrišče, a katero ni. Celo otroška igrišča ne zagotavljajo popolne

varnosti. Ni rešitev, da si zatiskamo oči pred napredkom telekomunikacij, ampak moramo le odgovorno ravnati in pustiti, da nas internet ne zasvoji in ne potegne v svoj skriti in nevarni svet.

Vir: Science illustrated, št. 62, januar 2015.



Z igranjem igre do razumevanja alzheimerjeve bolezni

Ian Franko Jesenšek, G 2. A

Pod okriljem nemškega Telekoma je podjetje za razvoj računalniških iger Glitchers za operacijska sistema Android in iOS ustvarilo brezplačno igro Sea Hero Quest, skozi katero pridobivajo pomembne podatke za razumevanje alzheimerjeve bolezni.

To je kronična nevrodegenerativna bolezen, ki predstavlja kar 70 % vseh znanih primerov demence. Njeni simptomi, ki vključujejo izgubo kratkotrajnega in dolgotrajnega spomina in izgubo orientacije v prostoru in času, se začnejo kazati že v starosti od 45 do 65 let. Leta 2016 je bolezen imelo kar 45 milijonov ljudi po celem svetu; znanstveniki predvidevajo, da bo do leta 2050 bolezen prizadela več kot 135 milijonov lju-

di. Do nedavnega ni bilo učinkovitega načina za raziskovanje bolezni. Zdaj je v pomoč posebna igra.

Na začetku igre igralec vpiše podatke, ki pripomorejo k raziskavi, kot so starost, spol in država, v kateri živi. Nato si mora zapomniti sliko zemljevida, ki opisuje pot do cilja. Ko prispe na cilj, izstreli signalno raketo nazaj proti začetku. Igra medtem shranjuje podatke o poti, izgubi smeri in drugo, pridobljene podatke pošlje v bazo podatkov, kjer čakajo na obdelavo.

Igro igra okoli 2.5 milijonov igralcev s celega sveta, ki so na ta način omogočili raziskovalcem neprecenljiv vpogled v razvoj bolezni, ki otežuje življenje mnogim starejšim ljudem.

Viri: <https://www.telekom.com/en/media/media-information/archive/deutsche-telekom-presents-innovative-new-mobile-game--sea-hero-quest—363198>, https://sl.wikipedia.org/wiki/Alzheimerjeva_bolezen#Epidemiologija, <http://www.bbc.com/news/technology-36203674>, <http://www.seaheroquest.com/en/gallery>, <http://www.alzheimers.net/resources/alzheimers-statistics/> (marec 2017).



Kaj sporočamo prek družbenih omrežij?

Esej za dijaško Gosarjevo nagrado 2017

(Fakulteta za uporabne družbene študije v Novi Gorici)

Matej Poljanšek, G 2. B

(Mentorica: Mojca Stare Aljaž, prof.)

Družbena omrežja so dandanes postala nepogrešljiv del našega vsakdana. Prek njih si izmenjujemo novice, klepetamo s prijatelji, izvemo, kaj je novega v svetu, ali pa preprosto pokažemo, kako se počutimo z objavo stilizirane fotografije. Virtualnim prijateljem, ki so lahko tudi resnični, povemo marsikaj, kar ne ustreza zlatemu pravilu, ki pravi, da lahko objaviš tiste stvari, ki bi jih tudi objavil na oglasni deski na javnem mestu, praviloma v šoli ali kje drugje, kjer objave lahko iznakazijo tvojo javno podobo. V glavnem mladi se tega pravila ne držimo, čeprav obstajajo tudi izjeme. Mladostniška neizkušenost nam prepreči, da bi racionalno odločali, katere objave lahko vplivajo na naše nadaljnje življenje, kot je na primer razgovor pri bodočem delodajalcu ali pa to, kaj si bodo mislili naši otroci, ko bodo iz otroške radovednosti iskali fotografije svojih staršev. Kot da to še ni dovolj, pa se pojavi tudi vprašanje nacionalnih in varnostnih služb, ki vsakodnevno zbirajo informacije, ki lahko predstavljajo grožnjo javni varnosti. Mi se niti ne zavedamo, kaj vse sporočamo organom pregona s svojimi selfiji in pogosto zgolj delno primernimi objavami na družbenih omrežjih. Hkrati pa se pojavi tudi vprašanje nadzora nad nami prek objav znanih oseb in drugih ljudi, ki imajo velik vpliv na javno mnenje.

Družbena omrežja so prvotno nastala z namenom ohranjanja in pri-

dobivanja stikov z znanci in ljudmi, ki smo jih spoznali na zabavi ali med potovanjem. S tako zamisljivo je bil ustanovljen Facebook. Njegov ustanovitelj, Mark Zuckerberg, ga je ustanovil z namenom ohranjanja stikov med študenti tudi po zabavah. Facebook je sedaj najmočnejše družbeno omrežje z več kot milijardo uporabnikov in prav toliko prenosov mobilne aplikacije na pametne naprave. Ljudje ga uporabljajo predvsem kot informacijsko orodje, saj ima vsaka kolikor toliko velika organizacija svojo osebno stran, kjer lahko izvemo osnovne informacije o organizaciji in tudi vse dejavnosti, ki jih je objavila na tako imenovanem osebni zidu. Poleg tega ga uporabljamo kot komunikacijsko orodje, sploh po tem, ko je izdal posebno mobilno aplikacijo, namenjeno izključno komunikaciji med uporabniki Facebooka.

Velikan med družbenimi omrežji je prav gotovo tudi Twitter. Omrežje, namenjeno predvsem krajšim besedilnim objavam, uporabljajo v največji meri politiki in druge osebe z velikim številom privržencev, saj lahko tako hitro in enostavno širijo svoje odzive na politično dogajanje in druge pomembne dogodke, ki jih sami s svojimi objavami naredijo pomembne. Med najvidnejšimi uporabniki Twitterja je zagotovo papež Frančišek, ki se tudi na ta način trudi približati mladim in vsem, ki se vsaj delno zanimajo za svet okoli sebe. Seveda pa pri omenjanju najvidnejših uporabnikov ne sme-

mo prezreti novoizvoljenega predsednika ZDA, Donalda Trumpa.

Da se ne osredotočimo le na klasična družbena omrežja, lahko kot primer netipičnega družbenega omrežja omenim YouTube, ki nam objave namesto prek besedil in fotografij predstavlja prek videoposnetkov in tako imenovanega life streaminga, ki neposredno prenaša video do uporabnika in mu s tem omogoča, da lahko v živo spremlja dogajanje. Na YouTubu lahko najdemo mnogo različnih vsebin, od glasbenih videospotov do amaterskih poljudnoznanstvenih razprav. Mnogi tako imenovani youtuberji širijo svoje mnenje in notranje informacije o novih posodobitvah iger in podobnih stvareh. YouTube je postal v zadnjem času pogosto uporabljen družbeni medij, saj ima vedno več ljudi dostop do hitrega interneta, ki pa je hkrati tudi glavna pomanjkljivost YouTubea, ker posnetki porabijo veliko količino podatkov, videoposnetki zavzamejo veliko več prostora kot klasični načini objav, torej besedilo in fotografije. To je razlog, da nima toliko uporabnikov na območjih s počasnejšo povezavo, kar pa je ponekod še danes zelo velika težava, ki jo poskuša rešiti celo Evropska komisija.

Poglejmo si še eno najnovejših družbenih omrežij, tako imenovani Snapchat. Gre za mobilno aplikacijo, prek katere si uporabniki izmenjujejo stilizirane fotografije, ki jih lahko posnamejo v sami aplikaciji, ki celo omogoča razne vizual-

ne nadgradnje fotografij. Lahko na primer posnamemo fotografijo, na kateri bomo izgledali, kot da imamo pasja ušesa in smrček. Seveda obstajajo zunanje aplikacije za podobno stilizacijo fotografij, a prek Snapchata lahko, kot nam tudi ime pove, komuniciramo z drugimi uporabniki. Ima pa tudi vgrajeno varovalo, ki preprečuje, da bi se fotografije predolgo prikazovale. Po določenem času se namreč skrijejo, kar preprečuje množično shranjevanje fotografij.

Vsa ta družbena omrežja so bila prvotno namenjena komunikaciji med uporabniki in deljenju njihovih prigod med njimi, a so do danes presegla ta namen. Mnogi, predvsem mladi z uporabo le-teh namreč poskušajo povečati svoj ugled v družbi in si s tem zagotoviti spoštovanje vrstnikov. To počnejo tudi z objavo neprimernih in velikokrat družbeno nesprejemljivih fotografij, česar pa uporabniki velikokrat sploh ne obsodijo, temveč to objavo ali fotografijo celo delijo naprej in tako zelo hitro zaokroži po internetu, s čimer postane neizbrisljiva in bo za vedno ostala rakrana za osebo, ki jo je prvotno objavila. V glavnem se mladi ne zavedajo, kaj v današnji dobi pomeni neizbrisljivo. Vsaka fotografija ali objava je dostopna širom sveta in posledično je pravzaprav zagotovljeno, da jo bo vsaj nekdo shranil na svoj računalnik in uporabil. To seveda ne pomeni, da je to v vsakem primeru slabo, lahko se najde



tudi neka oseba, ki je to storila z dobrimi nameni, a kaj, ko je takih veliko premalo oziroma skoraj nič, kar pa pomeni, da so pravzaprav nepomembni. Najbolj koristne fotografije za izsiljevanje so seveda tiste, na katerih je izsiljevanec v položaju, ki je družbeno nesprejemljiv ali celo popolnoma neprimeren ali pa je na neki fotografiji pomanjkljivo oblečen. To zadnje še posebej izkoriščajo izsiljevalci otrok. Ti se jim namreč prikupijo in dosežejo, da jim otroci brez vednosti staršev pošiljajo vedno bolj problematične fotografije samih sebe, čez čas pa od izsiljevanega zahtevajo nekaj v zameno, sicer bodo fotografije objavili in s tem nakopali sramoto tej osebi. Zanimivo pa je, da so mediji pravzaprav utihnili o teh zadevah, kar lahko razumemo kot zelo dober znak, ki pomeni, da se je število takih primerov spustilo skoraj na nič in so postali nezanimivi za medije. Tarča podobnih izsiljevalskih metod so velikokrat tudi znane osebnosti, katerih zgodbe in škandali ne ostanejo dolgo skriti. Lahko bi celo domnevali, da zvezdniki sami poskrbijo za take škandale le zato, da se o njih govori in da ljudje ne pozabijo nanje. Mnogi skoraj tekmujejo v škandalih in spornih objavah na

družbenih medijih. Družbene medije lahko izkoristimo tudi navadni smrtniki, če se hočemo igrati laične psihologe. Mnogo nam povedo o sebi njene objave, kar seveda izkoriščajo televizijski psihologi, specializirani prav v te namene raziskovanja osebnostnih profilov znanih posameznikov, katerih zgodbe lahko pridejo na naslovnice znanih časopisov in revij.

Seveda pa ne smemo pozabiti tudi na temnejše plati družbenih medijev, ki so lahko okno v naš notranji svet tudi ljudem, ki jim nočemo razkriti svojih osebnih informacij. Kot najznačilnejši primer lahko navedem kar varnostno-obveščevalne službe. Te morajo skrbeti za našo varnost in tudi za varnost države. V ta namen izkoriščajo navivnost ljudi pri objavljanju zaupnih informacij in predvsem njihovega političnega mnenja, kar jih lahko pripelje do suma, da je neka oseba nevarna za domovinsko varnost, četudi le izraža svojo jezo zaradi trenutne situacije glede objav v družbenih omrežjih. Prav tako izkoriščajo povezave med osebami na medmrežju za odkrivanje teroristov. Tako lahko pridejo do neke osebe zgolj po povezavah med pri

jateljji in niso niti najmanj sumljivi. Informacije, da tajne službe izkoriščajo družbena omrežja za nadzor, nam je razkril slavn ameriški žvižgač Edvard Snowden, ki je tudi ugotovil, da lahko povprečen CIA-heker dobi informacije iz naših osebnih računalnikov. Znanje vdiranja izkoriščajo tudi tatovi identitet. Po svetu mnogim ukradejo identiteto, vključno z dostopom do profilov družbenih omrežij, ki jih je žrtev uporabljala. Pred tem se lahko zavarujemo z močnimi gesli, ki nimajo nekega neposrednega namena, temveč potrebujemo nek ključ, da lahko ugotovimo njihov pomen. Pri drugih vrstah vdiranja se neprimerljivo težje zaščitimo. Največjo težavo pravzaprav predstavlja dejstvo, da mobilne aplikacije ne zahtevajo vedno novega vnosa ge-

sla, temveč je potrebna le enkratna prijava, zato lahko nekdo z vdorom v naš pametni telefon dobi dostop do vseh naših profilov, kar omogoča najmanj vpogled v našo zasebnost. K izboljšanju varnosti ne pripomorejo niti raziskave, ki dokazujejo, da je mogoče v pametni telefon vdreti tudi brez nekega izjemno dovršenega znanja, saj to zmorejo tudi navadni študenti. Najboljši način zavarovanja naših profilov ponuja podjetje Apple, ki nam omogoča nastavitve prstnega odtisa za odklepanje zaslona. Temu se dokaj hitro približujejo tudi drugi proizvajalci, le da njihovi bralniki prstnih odtisov še niso tako dovršeni in omogočajo obhod, kar je pokazalo tudi južnokorejsko tekmovanje, kjer so kitajski hekerji vdrlji v najnovejši Googlov telefon Pixel v zgolj minuti.



Naše objave so lahko močno orožje v boju proti nam samim, kar s pridom izkoriščajo mnogi, ki se želijo okoristiti na naš račun ali pa se preveč trudijo za varnost domovine, ne mislijo pa na varnost državljanov. Njim lahko naredimo veliko uslugo s tem, ko objavljamo sporne novice, ki smo si jih celo sami izmislili, in jim s tem damo razlog za izsiljevanje in ustrahovanje. Največja tarča smo seveda mladi, saj se imamo za računalniško najbolj pismene, hkrati pa pozabljamo, ko nekaj objavimo, da objavljeno ostane kljub izbrisu, zato se moramo zavedati, da nam lahko trenutna nepremišljenost uniči življenje, kar je škoda, saj ga imamo še kar nekaj pred sabo.

Kako v 10 korakih povečati varnost brezžičnih omrežij

Žan Jerič, R 3. A

Računalniška varnost in predvsem varnost brezžičnih omrežij sta nekaj, na kar uporabniki radi pozabijo. Še posebej tisti, ki še nimajo slabih izkušenj s kakršnimi koli vdori v svoje računalnike ali krajo gesel oz. podatkov. V tem članku je naštetih nekaj osnovnih korakov, ki lahko preprečijo vdor oz. krajo podatkov.

1. korak: Vključitev WPA (WPA2) ali WEP-kodiranja oz. enkripcije

Vsaka brezžična oprema omogoča vsaj en način kodiranja podatkov. Najbolje je seveda, da izberemo najboljše oz. najmočnejše kodiranje,

toda tega morata podpirati tako usmerjevalnik kot brezžična mrežna kartica pri sprejemniku. Zaradi tega je včasih potrebno izbrati nekoliko nižjo stopnjo kodiranja, kot jo omogoča usmerjevalnik.

2. korak: Zamenjava privzetega uporabniškega imena in gesla s svojim

Usmerjevalniki in dostopovne točke imajo ob nakupu tovarniško nastavljeno geslo, s katerim uporabnik dostopa do vseh pomembnih nastavitev. Ta privzeta uporabniška imena in gesla so po navadi enaka pri vseh napravah proizvajalca, zato napadalcu, ki ima vsaj

malo znanja, ni težko vdreti v omrežje. Pametno je izbrati kakšno naključno geslo z vsaj osmimi znaki.

3. korak: Filtriranje MAC-naslovov

Ta zaščita temelji na dejstvu, da ima vsak omrežni vmesnik (wi-fi, LAN ...) različno tovarniško nastavljeno heksadecimalno kodo. Tako lahko na usmerjevalniku nastavimo, da samo računalniki s točno določenim MAC-naslovom lahko dostopajo do wi-fi-omrežja. Če želimo pogledati MAC-naslov LAN-vmesnika na računalniku, gremo v cmd in vpišemo ukaz ip-

config /all, ki nam med drugim izpiše tudi fizični naslov (MAC) v obliki 00-A1-4B-56-77-B2. To se zdi idealna rešitev za vse zagate z brezžično varnostjo, saj bi teoretično na ta način lahko preprečili vsak nepooblaščen dostop. Žal pa vse ni tako enostavno. Že vsak malo boljši heker lahko namreč s posebnim programom ponaredi MAC-naslov in tako dostopi do omrežja. Problem te metode pa je v tem, da je potrebno MAC-naslove vpisovati ročno.

4. korak: Zamenjava imena brezžičnega omrežja (SSID)

Dostopovne točke in usmerjevalniki privzeto oddajajo ime omrežja, ki je po navadi drugačno pri vsakem proizvajalcu, a še vedno generično in enako za večino modelov. Če za primer vzamemo Linksys: SSID brezžičnega omrežja njihovih brezžičnih usmerjevalnikov je »linksys«. Napadalcem to zelo olajša delo, saj tako že iz imena razberejo, usmerjevalnik katerega proizvajalca ima uporabnik. Poleg tega pa je zanje to navadno znak, da je omrežje last uporabnika in da ni dobro zaščiteno.

5. korak: Onemogočanje oddajanja imena omrežja (SSID)

Dostopovne točke in usmerjevalniki tega navadno oddajajo v rednih časovnih presledkih. To je seveda nujno pri dostopovnih točkah, do katerih dostopa veliko različnih uporabnikov (npr. v lokalih). Ni pa to nujno pri domačih brezžičnih omrežjih, saj ko enkrat nastavimo vse odjemalce in jim ročno vpišemo ime omrežja, ti ne potrebujejo več SSID-sporočil. Ko jih enkrat ne oddajamo več, tudi močno zmanjšamo možnost, da bi se nekdo nepooblaščen hotel vključiti

v naše omrežje, saj to ne bo vidno na njegovem seznamu brezžičnih omrežij v dosegu. Spretnejši hekerji bodo še vedno znali zaznati vaše omrežje, toda kljub temu smo s tem korakom precej zmanjšali možnost vdora.

6. korak: Onemogočanje avtomatičnega povezovanja v odprta brezžična omrežja

Pomembno je tudi, da pri svojem računalniku ali pametnem telefonu onemogočite avtomatično povezovanje v odprta brezžična omrežja. Večina pametnih telefonov in računalnikov ima možnost avtomatičnega povezovanja v dosegljiva brezžična omrežja, ne da bi o tem obvestila uporabnika.

7. korak: Dodeljevanje statičnih IP-naslovov

Večina domačih upravljavcev omrežij uporablja dinamične IP--naslove prek DHCP-protokola. Ta je enostaven za postavitev, a žal ravno tako za napadalce, ki tako z lahkoto pridejo do veljavnih IP--naslovov. Da bi jim otežili delo v dostopovni točki ali usmerjevalniku, moramo izključiti DHCP in postaviti fiksno območje IP-naslovov. Zato raje uporabimo zasebne (privatne) IP-naslove, ki jih usmerjevalniki ne usmerjajo v internet in jih lahko vsak poljubno uporabi v svojem zasebnem omrežju (ne prihaja do konflikta z ostalimi omrežji z enakimi IP-naslovi). Da bodo te naprave lahko komunicirale z internetom, je potrebno naslove pretvoriti v javne naslove (za to na usmerjevalniku poskrbi NAT – Network Address Translation).

8. korak: Požarni zid

Večina usmerjevalnikov ima vgrajen požarni zid, ki ga ja mogoče

tudi izklopiti. Prepričajte se, da je vaš vklopljen. Ne zanašajte se le na tistega v vašem računalniku. Ne pozabite tudi na kakovosten protivirusni program, saj večina teh vključuje kvaliteten požarni zid.

9. korak: Varna in pravilna postavitev brezžične dostopovne točke ali usmerjevalnika

Kot varna oz. pravilna postavitev je mišljena postavitev na tako mesto, kjer bo čim manj signala uhajalo v okolico oz. izven stanovanja ali pisarne. Dlje kot bo segal signal, več ljudi ga bo lahko zaznalo in večja bo možnost vdora. To pomeni, da je dostopovne točke ali usmerjevalnike najbolje postaviti čim bolj na sredino stanovanja ali pisarne, in ne na okensko polico. S tem bomo izboljšali varnost, hkrati pa tudi poskrbeli za enakomerno razporeditev signala v svojih prostorih.

10. korak: Ko omrežja dlje časa ne potrebujete, ga lahko preprosto izklopite

(npr. ponoči ali medtem ko ste v službi ali na počitnicah). Ukrep je resnično enostaven, a tudi najbolj učinkovit in zanesljiv. Seveda to ne pomeni, da bi morali omrežje ugašati za vsakih 15 minut, ko ga ne uporabljate, saj bi bilo to nepraktično in moteče. Toda pri daljših intervalih to gotovo ne predstavlja večje motnje, sploh pa ne v primerjavi s prednostmi, ki jih ta preprost ukrep prinaša.

<http://www.racunalniske-novice.com/triki/piano-nasveti-za-povecanje-varnosti-brezicnih-omrezij.html>
<https://dronyx.wordpress.com/2008/05/14/zascita-domacega-wi-fi-omrezja/>

Podatkovne baze NoSQL

Aleksandar Lazarević, univ. dipl. inž.

Gašper Lomovšek, dipl. inž.

Razvoj poslovnih aplikacij, spletnih aplikacij in družbenih omrežij je nemogoče predstaviti brez uporabe SUPB (sistemov za upravljanje podatkovnih baz). Ti omogočajo razvoj aplikacij, ki zahtevajo shranjevanje, poizvedovanje, zagotavljanje integritete podatkov in kontrolo istočasnega pristopa do podatkov. Trenutno najbolj zastopani SUPB-ji temeljijo na relacijskih podatkovnih bazah in za upravljanje uporabljajo SQL-jezik [1].

Konvencionalni sistemi za shranjevanje senzorskih podatkov temeljijo na relacijskih podatkovnih bazah. Glede na rast števila senzor-

jev in meritev in potrebe po vse bolj zapletenih poizvedbah omejitve tradicionalnih relacijskih podatkovnih baz postajajo očitne. Pogledimo si naslednji scenarij: imamo 20 merilnih mest; na vsakem je 6 različnih senzorjev, ki vsakih 10 s izvajajo meritve in pošiljajo podatke v podatkovno bazo. Na dan to pomeni približno milijon meritev. Relacijska podatkovna baza z nekaj milijoni zapisov velja za veliko. Dodajanje novih senzorskih mest ali povečanje pogostosti zajema podatkov bistveno poveča število podatkov, ki jih je potrebno shraniti, in kompleksnost poizvedovanja po

shranjenih podatkih. Zato potrebujemo sistem, ki je razširljiv.

Kot rešitev naštetih problemov so se pojavile podatkovne baze NoSQL, ki prinašajo enostavno razširljivost v primerjavi s tradicionalnimi SUPB na velikem številu strežnikov in prilagojenost specifični uporabi, kot so družbena omrežja in rešitve v oblaku (angl. cloud computing). Te podatkovne baze niso relacijske in ponujajo možnosti dela z različnimi vrstami podatkov, kot so dokumenti ali grafi, in se zaradi tega imenujejo NoSQL – ne samo SQL (angl. Not only SQL) podatkovne baze [1].

Podatkovne baze NoSQL

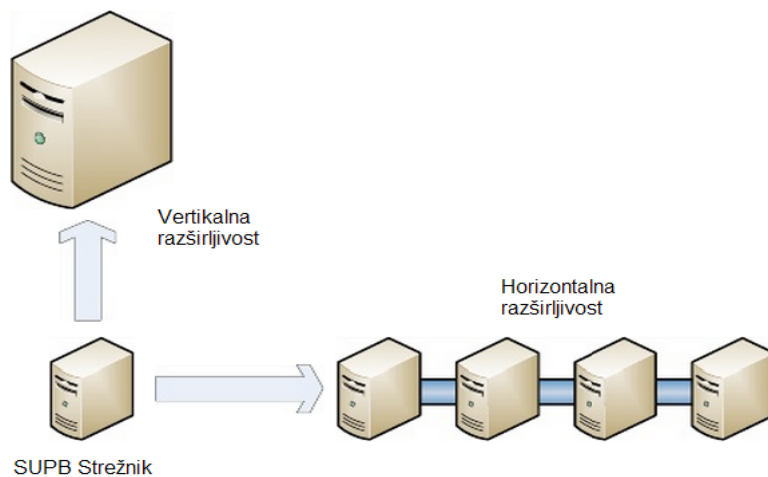
Specifičnost podatkovnih baz NoSQL

Največja prednost podatkovnih baz NoSQL je zagotovo horizontalna razširljivost. Povečane potrebe po virih se lahko zadovoljijo s povečanjem zmoglosti strežnika, na katerem se nahaja SUPB; to imenujemo vertikalna razširljivost. Ta je najlažja, ker ne zahteva spremembe podatkovne baze in/ali aplikacije. Vertikalna razširljivost je omejena z zmoglostmi strežnika. Druga oblika razširljivosti je dodajanje dodatnih strežnikov (angl. scaling out), na katere se razporeja delo; to imenujemo horizontalna razširljivost (glej sliko 1). Če preneha potreba po dodatnih strežnikih, te lahko odstranimo.

Da bi se omogočila enostavna horizontalna razširljivost (glede na relacijske SUPB), podatkovne baze NoSQL v glavnem ne zadovoljujejo ACID-pogojev. Čeprav se to

zdi resna težava, prav to omogoča glavne prednosti podatkovnih baz NoSQL: hitrost in izredna razširljivost. Upoštevanje ACID-pogojev povečuje kompleksnost SUPB. V določenih primerih so te lastnosti manj pomembne od hitrosti in raz-

širljivosti. Podatkovne baze NoSQL omogočajo replikacijo podatkov na več strežnikov s ciljem povečanja dostopnosti. Z izpadom enega ali več strežnikov SUPB lahko deluje še naprej brez prekinitev in izgube podatkov [2].



Slika 1: Horizontalna in vertikalna razširljivost (Vir: lastni)

CAP-izrek

CAP-izrek – znan tudi kot Brew-
rov izrek – je nastal leta 2000 in
pravi, da je nemogoče zadovoljiti
vse tri želene lastnosti porazdelje-
nih sistemov: konsistentnost (angl.
consistency), dostopnost (angl. ava-
ilability) in toleranco na izpad dela
sistema (angl. partition tolerance)
[2]. Pri razvoju SUPB se lahko iz-
bereta samo dve od navedenih treh
lastnosti porazdeljenih sistemov:

- konsistentnost (vse aktivno-
sti nad podatki, ne glede na
uspešnost, ohranijo podat-
kovno bazo v konsistentnem
stanju)
- dostopnost (vsak odjemalec
lahko vedno bere in piše v po-
datkovno bazo)
- toleranca na izpad dela sis-
tema (sistem lahko deluje ne
glede na to, da posamezni deli

sistema ne morejo več komu-
nicirati med seboj)

Tradicionalni SUPB su usmerjeni
v par lastnosti CA – konsistentnost
in dostopnost. Podatkovne baze
NoSQL so usmerjene v PA- ali PC-
par lastnosti. Tako se žrtvuje kon-
sistentnost ali dostopnost, obdrži se
pa toleranca na izpade, ki zagotavlja
enostavnejše doseganje razširljivosti.

Lastnosti in vrste podatkovnih baz NoSQL

Čeprav ne obstaja enostavna de-
finicija pojma NoSQL, je splošno
sprejeto, da so to podatkovne baze,
ki imajo naslednje lastnosti [3][4]:

- niso relacijske,
- podpirajo enostavno horizontal-
no razširljivost in replikacijo,
- podpirajo BASE-lastnosti (angl.
B - basically A - available S - soft
state E - eventual consistency) –
v nasprotju z ACID-lastnostmi
tradicionalnih relacijskih podat-
kovnih baz,

- rešitve so večinoma odprtoko-
dne,
- podpirajo različne načine shra-
njevanja podatkov.

BASE-lastnosti podatkovnih baz
NoSQL so:

- podatki so konsistentni, vendar
se ne more zagotoviti, da bodo
konsistentni takoj po vnosu
(angl. eventual consistency),
- podatki so spremenljivi (glede
na to, da se lahko spremenijo

tudi brez zunanjih vplivov zara-
di zagotavljanja konsistentnosti),

- podatki so dostopni tudi v pri-
merih, ko nekateri od vozlov
niso dostopni (angl. basically
available).

V nadaljevanju so predstavljeni
kratki opisi nekaj najbolj značil-
nih vrst podatkovnih baz NoSQL
glede na različne načine shranje-
vanja oz. strukturiranja podat-
kov [4].

Shrambe tipa *ključ-vrednost*

Podatkovna baza na osnovi parov
ključ-vrednost (angl. key-value) je naj-
bolj enostavna oblika podatkovne
baze. Prednosti takšnih baz so hitro
zapisovanje in branje podatkov in eno-

stavno doseganje razširljivosti ne glede
na količino podatkov. Pomanjkljivosti
so v zelo omejenih možnostih poizve-
dovanja in analize podatkov. Takšna
oblika podatkovne baze se uporablja

za shranjevanje ogromnih količin po-
datkov. Najbolj znani predstavniki so:
Redis, Riak, Valdemort, Berkley DB,
Oracle NoSQL, Amazon Dynamo,
Mem-cached, ...

Shrambe tipa *razširljiv zapis*

Podatkovne baze, ki temeljijo
na shrambi tipa *razširljiv zapis*
(angl. extensible record stores)

shranjujejo razširljive zapise, ki
so lahko vertikalno in horizon-
talno porazdeljeni po vozliščih.

Najbolj znani predstavniki so:
Cassandra, BigTable, Hbase,
Hypertable, ...

Shrambe na podlagi grafov

Podatkovne baze na podlagi gra-
fov temeljijo na strukturah, ki so
sestavljene iz vozlov in povezav
med njimi. Ne uporabljajo indek-

sov, ker ima vsak vozle povezavo
do sosednjih. Ti in povezave imajo
lahko večje število različnih atri-
butov. Uporabne so za shranje-

vanje podatkov, kot so družabna
omrežja ali geografske lokacije.
Najbolj znani predstavniki so: Ne-
o4J, InfoGrid in IMS.

Shrambe na podlagi dokumentov

Najpogosteje uporabljane podatkovne baze NoSQL temeljijo na dokumentih (angl. document oriented). Vsak podatek je en dokument, katerega struktura ni vnaprej določena, kar pomeni, da imajo različni dokumenti lahko različne attribute. Vsak

dokument ima enolični ključ, ki se uporablja za poizvedovanje in izmenjavo dokumentov. Glede na to, da ni potrebno vnaprej določiti sheme podatkov, se le-ti pripravijo tako, kot se z njimi dela v aplikacijah oz. jih ni potrebno normalizirati. Takšen pristop

olajša razvoj aplikacij, kjer vnaprej ni znano, s kakšnimi podatki se bo delalo, in kjer je potrebno shranjevati več podatkov znotraj neke skupine.

Najbolj znana predstavnik sta: CouchDB in MongoDB.

MongoDB

MongoDB je dokumentna NoSQL-podatkovna baza, katere razvoj se je začel leta 2007 v podjetju 10gen [5]. Podatkovna baza je sestavljena iz zbirk (angl. collection). Ta vsebuje več dokumentov (glej sliko 2).

Zbirke in baze se ustvarijo takrat, ko se doda prvi dokument (angl. lazy). Dokument je množica parov ključ-vrednost, kjer je ključ naziv (niz znakov – string), vrednost pa eno od naslednjega:

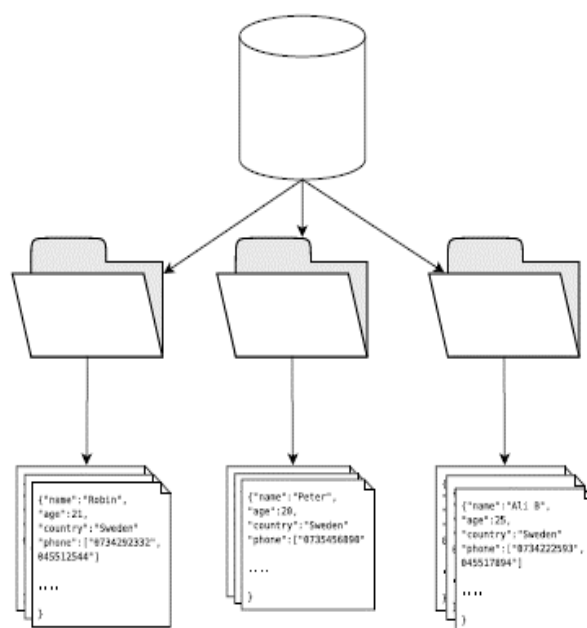
- osnovni tip podatka (string, integer, timestamp, ...),
- vgnazdeni dokument ali
- polje vrednosti (angl. array).

Obstaja še posebni par ključ-vrednost, in sicer `_objectid`, ki predstavlja privzeti identifikator dokumenta. Dokumenti v isti zbirki imajo lahko različne strukture. Vsak se fizično shrani kot en BSON-objekt. BSON je binarna oblika JSON-objekta, JSON (angl. JavaScript Object Notation) pa je format zapisovanja podatkov v obliki JavaScript sintakse, ki je razvit zaradi enostavnega branja, iskanja in razčlenjevanja, zapis pa zavzame zelo malo prostora. MongoDB podpira indeksiranje po katerem koli atributu dokumenta. Indeksi so implementirani kot B-drevesa in drastično povečajo performanse poizvedb. MongoDB ima

Baza

Zbirke

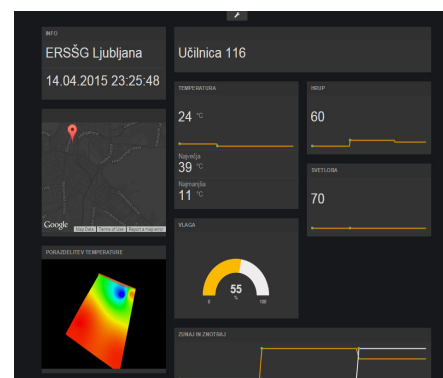
Dokumenti



Slika 2: Struktura baze podatkov v MongoDB (Vir: lastni)

vgrajeno podporo za repliciranja baze, in sicer v različnih oblikah (single master/single slave, multiple master/multiple slave, master/master, ...) [6].

Na šoli je bil narejen in predstavljen prototip sistema za zajem in vizualizacijo okoljskih podatkov, ki je uporabljal dokumentno podatkovno bazo MongoDB. Ta se je izkazala za zelo hitro in prilagodljivo, kar je omogočilo hitro poizvedovanje, to pa je pogoj za sprotno vizualizacijo.



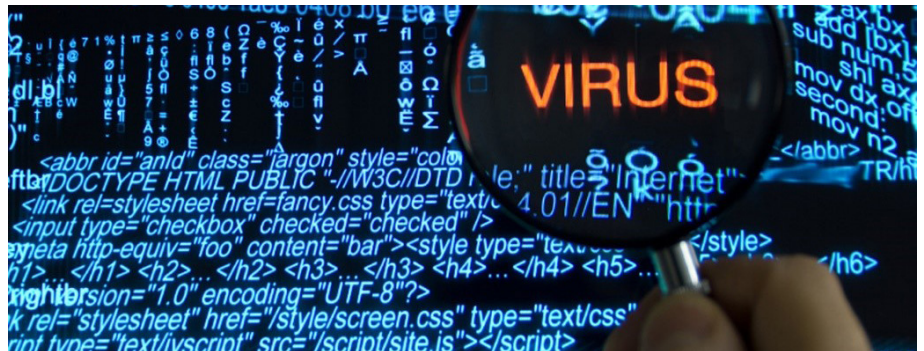
Slika 3: Nadzorna plošča (angl. dashboard) – vizualizacija okoljskih podatkov v učilnici 116 (Vir: lastni)

Računalniški virusi in zaščita pred njimi

Jernej Toth, G 2. B

Literatura:

- [1] STRAUCH C., (2010): *Selected Topics on Software Technology – Ultra-Large Scale Sites*, Hochschule der medien, Stuttgart.
- [2] KNEZ N., (2011): *Primerjava relacijske in NoSQL podatkovne baze in opredelitev kriterijev za pomoč pri izbiri najprimernejše podatkovne baze*. Diplomsko delo univerzitetnega študija, Fakulteta za računalništvo in informatiko, Univerza v Ljubljani.
- [3] ŠKERJANC N., (2012): *Primerjava učinkovitosti NoSQL in relacijske podatkovne baze*. Diplomsko delo, Fakulteta za računalništvo in informatiko, Univerza v Ljubljani.
- [4] NATARAJAN, R. (2010): *A Survey Report on databases designed for Cloud*, [web paper] http://blogs.oracle.com/natarajan/entry/a_survey_report_on_nosql (Dostop: 1. 2. 2014).
- [5] MongoDB (2014): *MongoDB manual 2.4*, [web paper].
- [6] <http://docs.mongodb.org/manual/> (Dostop: 1. 2. 2014).



Kaj sploh je računalniški virus?

Pomislimo na biološki virus, tak, ki nas okuži z neko boleznijo. Ko smo okuženi, naše telo ne deluje popolnoma pravilno. Računalniški virus je zelo podoben biološkemu. Naš računalnik okuži tako, da programi ne delujejo pravilno, lahko nam zbrše določene datoteke ali pa celo onemogoči pravilen zagon računalnika ob naslednjih poskusih vklopa. Na internetu je danes več kot deset tisoč virusov in vsak dan odkrijejo še kakšnega novega.

Kako računalnik sploh okužimo z virusom?

Kljub izjemni pozornosti, da ne bi dobili virusa, se to lahko zgodi. Virus lahko dobimo že pri povsem vsakdanjih opravilih na internetu, kot so deljenje glasbe, filmov ali slik z ostalimi, ogled spletne strani, okužene z virusom, odpiranje e-pošte, ki vsebuje virus in tako dalje.

Kaj nam virus lahko stori?

Nekateri virusi so sprogramirani tako, da nam poškodujejo programe, različne vrste datotek ali pa nam popolnoma ponastavijo trdi disk. Poznamo tudi viruse, ki naše omrežje tako obremenijo, da je

nesposobno opraviti kakršno koli zahtevo, ki potrebuje internetni dostop za izvršitev. Tisti najmanj škodljivi pa le očitno upočasnijo našo napravo ali pa povzročijo več nenadnih sesutij.

Kako ugotovimo, da je računalnik okužen?

Če imamo napravo okuženo z virusom, bomo pri njej opazili, da je počasnejša kot po navadi, nenadno izgubo datotek, pogoste ponovne zagone ali pa se bo naša naprava obnašala nepredvidljivo in ne v skladu z našimi zahtevami.

Kako se lahko zaščitimo pred virusom?

Ko se enkrat spoznamo z možnostmi okužbe svojega računalnika z virusom, se lahko izognemo več virusom in takšnim in drugačnim vabam. Da pa svoj računalnik maksimalno zavarujemo, imamo podjetja, ki za nas uporabnike proizvajajo antivirusne programe. Pred virusi se lahko zaščitimo tudi z rednim posodabljanjem operacijskega sistema, antivirusnih programov, povečamo internetno zaščito in se izogibamo sumljivim spletnim stranem.

NGINX

Tadej Gašparovič, R 1. C
(Mentor: Aleš Volčini, prof.)

NGINX je brezplačen, odprto koden, visoko zmogljiv HTTP-strežnik in vzvratni posrednik (angl. reverse proxy), IMAP/POP3-posredniški strežnik (angl. proxy server) in generičen TCP/UDP-posredniški strežnik (angl. generic TCP/UDP proxy server), ki ga je originalno napisal Igor Sysoev.

Znan je po visoki zmogljivosti, stabilnosti, bogatem naboru funkcij, enostavni konfiguraciji in nizkih sistemskih zahtevah. Dolgo časa je poganjal veliko zelo obremenjenih ruskih spletnih strani, kot so Yandex, Mail.Ru, VK in Rambler. Po podatkih Netcrafta je NGINX v septembru 2016 postregel ali pos-

redoval 27,6 % najbolj zaposlenih spletnih strani. Nekatere bolj poznane, v ozadju katerih je NGINX, so Netflix, WordPress.com, Hulu, Pinterest, CloudFlare, GitHub, SoundCloud in veliko drugih.

NGINX je eden izmed redkih strežnikov, napisanih za reševanja C10K-problema (strežnik deluje brez težav tudi pri velikem številu vzpostavljenih povezav). V primerjavi s tradicionalnimi strežniki se NGINX ne uporablja niti (angl. threads) za procesiranje zahtev. Namesto tega uporablja veliko bolj razširljivo, asinhrono arhitekturo, ki temelji na dogodkih (angl. asynchronous event-based archi-



tecture). Ta arhitektura uporablja majhne, vendar predvidljive količine pomnilnika pod obremenitvijo. Tudi če ne pričakujete, da boste procesirali tisoče zahtev naenkrat, sta vam visoka zmogljivost in majhna poraba pomnilnika lahko še vedno v korist. NGINX je razširljiv v vseh pogledih: od najmanjšega virtualnega privatnega strežnika (angl. virtual private server) pa vse do velikih cluster računalnikov in strežnikov.

Zgodovina NGINX-a

Igor Sysoev je pričel z razvojem NGINX-a leta 2002. Originalno je bil NGINX razvit za potrebe spletnih strani, kot je Rambler, ki je do septembra 2008 stregel tudi do 500 milijonov strani na dan.

Julija 2011 je Igor Sysoev ustanovil podjetje z istim imenom za zagotavljanje komercialnih izdelkov in podpore za programsko opremo NGINX. Sedež podjetja je v San Franciscu v Kaliforniji. Oktobra 2011 je NGINX zbral 3 milijone ameriških dolarjev od BV Capital, Runa Capital in MSD Capital. Podjetje je napovedalo komercialno podporo za firme, ki uporabljajo NGINX v produkciji. To so napovedali februarja 2012, plačljivo NGINX Plus naročnino pa v avgustu 2013. Pri podpornih paketih so pomembne namestitve, konfigu-

racija, izboljšava zmogljivosti itd. Podpora vključuje proaktivna obvestila o večjih spremembah, varnostnih popravkih (angl. security patches), nadgradnjah in popravkih (angl. patches). Automattic, razvijalci spletne strani WordPress.com, in MaxCDN, ponudnik sistema za

dostavljanje vsebine (angl. content delivery system provider), sta postala partnerja za financiranje nadgradnje Googlovega SPDY verzije 3.1, načrtovane za začetek leta 2014. NGINX nudi tudi svetovanje za pomoč strankam pri konfiguraciji po meri in dodajanje novih funkcij.



Namestitev

Postopek namestitve NGINX-strežnika je odvisen od operacijskega sistema.

Minimalne zahteve strojne opreme

Glede na to, da NGINX uporablja asinhrono arhitekturo, ki temelji

na dogodkih, so sistemske zahteve zelo majhne. Uradna spletna stran NGINX.org ne da točnega podatka,

vedar na podlagi mnenja izkušenih uporabnikov lahko sklepam, da so minimalne zahteve naslednje:

CPU	RAM	Prostor na disku
2 jedra, 2,0 GHz	128 MB	2 MB+

Minimalne zahteve programske opreme

NGINX ravno tako kot pri strojni tudi pri programski opremi ni preveč zahteven. Brez težav ga lahko namestimo na naslednjih operacijskih sistemih (sicer je programsko opremo potrebno prevesti iz izvirne kode):

Operacijski sistem	Različice operacijskega sistema
Ubuntu	12.04 LTS (i386, x86_64) 14.04 LTS (i386, x86_64, aarch64/arm64) 16.04 LTS (i386, x86_64, ppc64le)
CentOS	5.x, 6.x (i386, x86_64) 7.x (x86_64)
Debian	7.x, 8.x (i386, x86_64)
Red Hat	5.x, 6.x (i386, x86_64) 7.x (x86_64)
SUSE Linux Enterprise Server	12 (x86_64)

Namestitev na linuxu

Za namestitev NGINX-strežnika na linuxu se lahko uporabijo namestitveni paketi s spletne strani nginx.org oz. Na večini linux distribucijah lahko NGINX-strežnik preprosto namestimo s pripomočkom za upravljanje paketov (angl. package management utility) apt-get.

```
root@debianTEST2VM:~# apt-get install nginx
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following extra packages will be installed:
  nginx-common nginx-full
Suggested packages:
  fcgiwrap nginx-doc ssl-cert
The following NEW packages will be installed:
  nginx nginx-common nginx-full
0 upgraded, 3 newly installed, 0 to remove and 0 not upgraded.
Need to get 589 kB of archives.
After this operation, 1,410 kB of additional disk space will be used.
Do you want to continue? [Y/n]
```

Namestitev NGINX na linuxu

Namestitev na FreeBSD

Na FreeBSD lahko NGINX namestimo iz paketov ali skozi ports sistem. Ta nudi večjo fleksibilnost, kar omogoča izbiro med širokim izborom funkcij. Port bo prevedel NGINX z izbranimi funkcijami in ga namestil.

Prevajanje izvirne kode

Če se potrebuje posebna funkcionalnost, se lahko NGINX prevede tudi iz izvirne kode. Ta pristop je zelo fleksibilen, a je za začetnika lahko dokaj zapleten.

```
2  /*
3  * Copyright (C) Igor Sysoev
4  * Copyright (C) Nginx, Inc.
5  */
6
7
8  #include <ngx_config.h>
9  #include <ngx_core.h>
10 #include <nginx.h>
11
12
13 static void ngx_show_version_info(void);
14 static ngx_int_t ngx_add_inherited_sockets(n
15 static ngx_int_t ngx_get_options(int argc, c
16 static ngx_int_t ngx_process_options(ngx_cyc
17 static ngx_int_t ngx_save_argv(ngx_cycle_t *
18 static void *ngx_core_module_create_conf(ngx
19 static char *ngx_core_module_init_conf(ngx_c
20 static char *ngx_set_user(ngx_conf_t *cf, ng
21 static char *ngx_set_env(ngx_conf_t *cf, ngx
22 static char *ngx_set_priority(ngx_conf_t *cf
23 static char *ngx_set_cpu_affinity(ngx_conf_t
24 void *conf);
25 static char *ngx_set_worker_processes(ngx_co
26 void *conf);
27 static char *ngx_load_module(ngx_conf_t *cf,
28 #if (NGX_HAVE_DLOPEN)
29 static void ngx_unload_module(void *data);
30 #endif
31
32
33 static ngx_conf_enum_t ngx_debug_points[] =
34 { ngx_string("stop"), NGX_DEBUG_POINTS_S
35 { ngx_string("abort"), NGX_DEBUG_POINTS_
36 { ngx_null_string, 0 }
37 };
38
39
```

NGINX-izvirna koda

Konfiguracija

NGINX-konfiguracijska datoteka (*nginx.conf*) se na večini sistemov nahaja v */etc/nginx/*, lahko pa tudi v */opt/nginx/conf/*.

Struktura konfiguracijske datoteke

NGINX je sestavljen iz modulov, ki jih nadzirajo direktive v konfiguracijski datoteki. Direktive delimo na enostavne (angl. *simple directives*) in blokovne (angl. *block directives*). Enostavno direktivo sestavlja ime in parameter direktive, ločena s presledkom in konča se s podpičjem (;). Blokovna direktiva ima enako strukturo, le da se konča z zavrtima oklepajema ({ in }), ki vsebujeta dodatna navodila. Če lahko blok direktiva vsebuje druge direktive znotraj zavrtih oklepajev, se imenuje kontekst (angl. *context*; npr. *events*, *http*, *server*, *location*). Direktive, ki se ne nahajajo znotraj konteksta, so zajete znotraj glavnega konteksta (angl. *main context*). Vedno pa velja, da se *events* in *http*-direktive nahajajo znotraj glavnega konteksta, *server* znotraj *http*-konteksta in *location* znotraj *server*-konteksta.

Preostanek vrstice za znakom # se upošteva kot komentar.

Streženje statične vsebine

Pomembna naloga spletnih strežnikov je streženje datotek, kot so slike in statični HTML-dokumenti. V naslednjem primeru je strežnik konfiguriran, da glede na zahtevo postreže datoteke iz različnih lokalnih poti: */data/www* (lahko vsebuje HTML-datoteke) in */data/images* (vsebuje slike). Da dosežemo želeno, moramo urediti konfiguracijsko datoteko (*nginx.conf*) in nastaviti *server* blok znot-

raj *http*-bloka z dvema *location* blokom.

Najprej ustvarimo obe mapi in vanje shranimo potrebne datoteke (*index.html* v */data/www* in nekaj slik v */data/images*). Nato odpremo konfiguracijsko datoteko in zakomentiramo vse primere *server* bloka. Znotraj *http*-bloka lahko ustvarimo nov *server* blok.

```
http {
    server {
```

Konfiguracijska datoteka lahko vsebuje več *server* blokov, ki se razlikujejo po vratih (angl. *port*), na katerih poslušajo, in imenih (angl. *server names*). Ko se NGINX odloči, kateri *server* blok procesira določeno zahtevo, primerja URI-zahteve s parametri *location* direktiv, definiranih znotraj *server* bloka.

V *server* blok dodamo *location* blok:

```
location / {
    root /data/www;
```

Blok *location* določi lokalno lokacijo, kjer se nahajajo datoteke za zahtevano lokacijo »/«. Če se zahteva ujema, NGINX doda zahtevan URI poti, določeni v *root* direktivi, in tako tvori lokacijo datoteke na lokalnem sistemu (torej za zahtevan URI »/index.html« dobimo lokalno pot »/data/www/index.html«). Če se ujema več *location* blokov, potem NGINX izbere blok z najdaljšo predpono. *Location* blok zgoraj poda

najkrajšo možno predpono dolžine ena, torej če se noben blok ne ujema, je uporabljen ta blok.

Dodamo še drug *location* blok:

```
location /images/ {
    root /data;
}
```

Ta blok se ujema z vsemi zahtevami, ki se začnejo z */images/* (*location /*, se ravno tako ujema s takimi zahtevami, ampak ima krajšo predpono.

Končni *server* blok izgleda tako:

```
server {
    location / {
        root /data/www;
    }

    location /images/ {
        root /data;
    }
}
```

To je že delujoča konfiguracija NGINX-strežnika, ki posluša na privzetih vratih 80, do njega lahko dostopamo na <http://localhost/> (na lokalni napravi).

Vzpostavitev enostavnega posredniškega strežnika

NGINX je pogosto uporabljen kot posredniški strežnik, kar pomeni, da NGINX posreduje vse zahteve »posredovanim« strežnikom (angl. *proxied servers*), pridobi odgovore in jih posreduje klientom.

V nadaljevanju je predstavljen primer konfiguracije takega strežnika. V primeru sta uporabljena dva strežnika, definirana na isti NGINX-in-

Welcome to nginx on Debian!

If you see this page, the nginx web server is successfully installed and working on Debian. Further configuration is required.

For online documentation and support please refer to nginx.org

Please use the `reportbug` tool to report bugs in the nginx package with Debian. However, check [existing bug reports](#) before reporting a new bug.

Thank you for using debian and nginx.

NGINX-pozdravni zaslon

stanci (angl. instance). Prvi se bo obnašal kot posrednik in bo posredoval vse zahteve (razen zahteve za slikovne datoteke) drugemu strežniku. Zahteve za slikovne datoteke pa bo obdelal sam (tj. posredniški strežnik) in postregel ustrezno datoteko iz lokalne shrambe.

Najprej definiramo posredniški strežnik, tako da konfiguraciji dodamo še en server blok, kot sledi:

```
server {
    listen 8080;
    root /data/up1;
    location / {
    }
}
```

To bo enostaven strežnik, ki posluša na vratih 8080, kot smo določili z `listen` direktivo (ta direktiva prej ni bila uporabljena, saj smo uporabljali privzeta/standardna vrata 80). Strežnik bo za vse zahteve uporabil pot na lokalnem sistemu `/data/up1` (ta direktorij moramo tudi ustvariti tako kot `index.html` na isti poti). Upoštevajte, da se `root` direktiva nahaja znotraj `server` konteksta, in ne znotraj `location` konteksta kot v prejšnjem primeru. Taka `root` direktiva je uporabljena, ko `location` blok, izbran za streženje, ne vsebuje le-te.

Sedaj uporabimo server konfiguracijo iz prejšnjega primera in jo prilagodimo, da se obnaša kot posredniški strežnik. Znotraj prvega `location` bloka uvedemo `proxy_pass` direktivo s protokolom, imenom in vrati »posredovanega« strežnika (v našem primeru <http://localhost:8080>).

```
server {
    location / {
        proxy_pass http://
localhost:8080;
    }
    location /images / {
        root /data;
    }
}
```

Drugi `location` blok, ki trenutno postreže datoteke iz poti `/data/images` za vse zahteve s predpono `/images/`, bomo prilagodili, da primerja zahteve za slikovne datoteke s končnicami datotek, ki so tipične za datoteke te vrste.

```
location ~
\.(gif|jpg|png)$ {
    root /data/images;
}
```

Parameter je regularni izraz (angl. regular expression), ki se ujema z vsemi URI-ji, ki se končajo z `.gif`,

`.jpg` ali `.png`. Regularne izraze se predznači s `~`. Za vse zahteve, ki se ujemajo s tem izrazom, so datoteke postrežene iz poti `/data/images`.

Ko NGINX izbira `location` direktivo, ki bo uporabljena za streženje, najprej preveri direktive, ki določajo predpone, in si zapomni direktivo z najdaljšo predpono, nato pa regularne izraze. Če se katera od regularnih izrazov ujema z zahtevo, potem NGINX izbere ta `location` blok za streženje, sicer pa je uporabljen `location` blok z najdaljšo predpono, ki si ga je NGINX zapomnil pred preverjanjem regularnih izrazov.

Končna konfiguracija NGINX-a izgleda, kot sledi:

```
server {
    location / {
        proxy_pass http://
localhost:8080;
    }
    location ~
\.(gif|jpg|png)$ {
        root /data/images;
    }
}
```

Ta strežnik bo filtriral vse zahteve, ki se končajo z `.gif`, `.jpg` ali `.png`, in te datoteke postregel iz lokalne shrambe `/data/images` in posredoval vse ostale zahteve »posredovanemu« strežniku, konfiguriranemu zgoraj.

Za uveljavitev sprememb pošljemo `reload` signal NGINX-procesu z ukazom `nginx -s reload`.

Zaključek

Iz vsega napisanega lahko sklepamo, da je NGINX zelo lahek (angl. lightweight), vendar močan strežniški program. Podpira vse osnovne funkcije http-strežnika, kot so streženje statičnih datotek in streženje dinamičnih datotek / strani (integracija s skriptami, kot sta PHP, Python itd.), podpira pa tudi nekatere naprednejše funkcije, kot je konfiguracija virtualnih strežnikov.

INSTANCE
1-4

Annual subscription for individual licenses

- Choose your support level
- Get up and running in minutes

[BUY ONLINE](#)

VOLUME
5+

Volume pricing for 5+ instances

- Developer Licenses available
- Ideal for small clusters and high-availability

[CONTACT SALES](#)

APPLICATION
∞

Unlimited instances & full support

- Ideal for quickly growing apps
- Deploy with containers and microservices

[CONTACT SALES](#)

Ponudba NGINX Plus

Velike prednosti NGINX-a so njegova fleksibilnost in izjemno majhne zahteve za strojno opremo, zahvaljujoč njegovi asinhroni arhitekturi.

V tej seminarski nalogi sem se osredotočil predvsem na http-strežnik, vendar NGINX ponuja še veliko drugih strežnikov, kot je generičen TCP/UDP-strežnik. Ravno tako

sem se osredotočil na brezplačne funkcije NGINX-a, vendar NGINX Inc. nudi tudi plačljivo različico, NGINX Plus, za katero nudijo tudi podporo.

Viri:

Beginner's Guide. 2016. NGINX [27. 10. 2016].

Dostopno na spletnem naslovu: http://nginx.org/en/docs/beginners_guide.html

Nginx. 2016. Wikipedia, the free encyclopedia. [27. 10. 2016]. Dostopno na spletnem naslovu: <https://en.wikipedia.org/wiki/Nginx>

Igor Sysoev. 2016. Wikipedia, the free encyclopedia. [27. 10. 2016]. Dostopno na spletnem naslovu: https://en.wikipedia.org/wiki/Igor_Sysoev

NGINX Pricing. 2016. NGINX [27. 10. 2016]. Dostopno na spletnem naslovu: <https://www.nginx.com/products/pricing/>

Kako je umetna inteligenca postala del našega vsakdana

Andrej Marsetič, R 3. B

Samoučeča se tehnologija in strojno učenje predstavljata del zelo popularne teme današnjih dni – umetne inteligence. Skupaj predstavljata naslednjo generacijo aplikacijskega razvoja in upravljanja.

Kaj je umetna inteligenca?

Umetna inteligenca je močno prepletena s psihologijo, nevrologijo, matematiko, logiko, filozofijo in z drugimi vedami. Od progra-

merja zahteva poznavanje vseh navedenih področij. Koda takega programa je seveda zelo zapletena. Programer mora podrobno poznati način razmišljanja in čustvovanja telesa, ki ga bo program simuliral.

Njeno delovanje je v osnovi zelo preprosto. Sistem zbira podatke iz različnih podsistemov (termometer, kamera, pogosto vnesene be-

sede ...), jih shrani in analizira. Sčasoma, z uporabo tega cikličnega sistema, lahko računalnik spremeniti svoj model tako, da ustreza danim okoliščinam.

Umetna inteligenca je v zadnjih letih zelo napredovala. Tako je na nekaterih področjih že prehitela človeški um. IBM-ova Globoka modrina (Deep Blue) je človeka pri šahu prvič premagala že pred sko-

raj 20 leti. Enako usodo je doživela večina miselnih iger.

Zelo zanimivo pa je dejstvo, da je več tisoč let staro azijsko igro Go računalnik (Google DeepMind) premagal šele pred kratim. AlphaGo je program podjetja Google DeepMind, ki simulira nevtronska omrežja v možganih, ta deluje s pomočjo vzpostavljanja medsebojnih povezav. V osnovi je bil opremljen le s pravili igre, vse ostalo se je naučil sam s pomočjo svojih soigralcev – ljudi.

Tudi veliki proizvajalci pametnih naprav že imajo svoje samoučeče programe. Siri, Cortana, Google Assistant in Alexa so štiri rešitve strojnega učenja štirih velikanih (Apple, Microsoft, Google in Amazon). Vse so v osnovi enako funkcionalne. Uporabnik jim glasovno ali tekstovno sporoči, kaj morajo storiti (npr. poklicati Deja-

na, naročiti hrano, ustvariti opomnik ...).

Umetna inteligenca oz. semantična tehnologija se lahko izkaže za zelo učinkovito pri problemih, ki jih rešujemo z relacijskimi bazami, torej pri poizvedbah po svetovnem spletu. Semantično tehnologijo že nekaj let uporabljajo Google, Hacking pa tudi domači Najdi.si.

Svetla prihodnost

Umetna inteligenca se razvija s svetlobno hitrostjo. Uporabljali jo bomo lahko na vseh področjih. Računalniki bodo npr. znali analizirati bolezen in nam predpisati recept, oglaševalci bodo še bolje poznali naše navade in nam znali ob pravem trenutku ponuditi stvari, ki jih potrebujemo. V nastajanju je že nov velik projekt The Human Brain Project, ki bo simuliral sintetične možgane, podobne našim biološkim. Ob vsem

tem napredku pa se sprašujem, ali bo sledila velika vojna, vojna med računalnikom in človekom.

Viri:

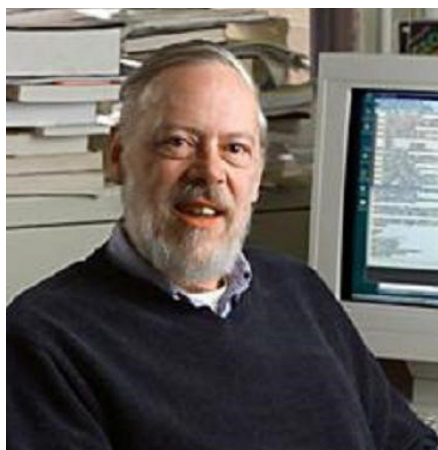
Nichols, Greg. 2015. *Self-learning systems to replace humans in manufacturing*. ZDNet[online]. 28. april 2015. [Citirano 20. jan. 2017]. Dostopno na spletnem naslovu: < <http://www.zdnet.com/article/self-learning-systems-to-replace-humans-in-manufacturing/> > .
Huš, Matej. 2015. V šahu čedalje več remijev, prednost belega pa ostaja. Slo-tech[online]. 10. apr. 2015. [Citirano 1. feb. 2017]. Dostopno na spletnem naslovu: <https://slo-tech.com/novice/t641272#crt>
Huš, Matej. 2016. AlphaGo po treh zmagah že skupni zmagovalec dvoboja v goju proti človeku. Slo-tech[online]. 12. mar. 2016. [Citirano 1. feb. 2017]. Dostopno na spletnem naslovu: <https://slo-tech.com/novice/t668972#crt>

Računalniški polbogovi

Darjan Toth, dipl. inž.

Polbog je mitološko bitje, ki ima človeške in božanske lastnosti. Gre torej za heroje z nekimi izjemnimi lastnostmi, gre za vzornike, za osebe, ki jih spoštujemo in običajno zelo častimo.

Če bi povprečnega zemljana povprašal, kdo je zanj računalniški polbog, bi velika večina znala naštet dve imeni: Bill Gates in Steve Jobs. Verjetno bi večina mlade generacije vegovcev na ta seznam dodala še Marka Zuckerberga, potem pa bi se ta seznam počasi končal. Sem pa povsem prepričan, da skoraj nihče ne bi omenil imena Dennis Ritchie.



Steve Jobs je umrl 5. oktobra 2011. Novice so poročale o izgubi resnično velikega uma, pokopan je bil z vsemi častmi, za njim je žalovalo več milijonov

ljudi po vsem svetu. Le teden dni kasneje, približno 12. oktobra, je umrl Dennis Ritchie. Za njim niso žalovale cele množice oboževalcev, novica o njegovi smrti ni bila objavljena nikjer v medijih, še več, datum njegove smrti sploh ni znan, saj so ga 12. oktobra našli mrtvega v njegovem domu v Berkeley Heights v New Jerseyju, kjer je živel sam. Priznam, tudi sam nisem vedel za njegovo smrt, vse dokler mi ni dijak 2. letnika med glavnim odmorom dejal: »Profesor, a ni to tisti Dennis, kterga smo nekaj omenjali na začetku leta? Tuki na netu piše, da je kao umrl ...«

Po vsem napisanem bi lahko bralec mirne volje sklepal in zaključil, da Dennis Ritchie že ni mogel biti tako pomemben za razvoj računalništva, da bi ga bilo vredno omenjati v tem članku. Kaj šele, da bi ga enačili z že prej omenjenimi računalniškimi polbogovi in velikani. A bi se motil. Močno motil. Dr. Dennis MacAlistair Ritchie ni bil čudežni, več milijard dolarjev vreden deček iz Silicijeve doline, ki bi imel navado z velikega, dvignjenega odra v minimalističnem črnem puliju očarati množice s svojimi svetlečimi novimi izdelki in hkrati z divjo retoriko zmesti konkurenco. Prav nasprotno, Ritchie je bil bolj bradat, zelo skromen, tih, nekoliko neurejen računalničar, ki je nosil jopico in puloverje in imel razmetano pisarno. Pa vendar je bil računalniški znanstvenik v pravem pomenu besede.

Rodil se je leta 1941 in leta 1960 je že diplomiral iz fizike in uporabne matematike na Harvardu in se po sledih svojega očeta zaposlil v Bellovih laboratorijih, ki so takrat predstavljali žarišče tehnološkega razvoja v Združenih državah Amerike. Leta 1968 je Ritchie doktoriral in v letih od 1969 do 1973 razvil svoj programski jezik C. Uporabil ga je za prenovo operacijskega sistema Unix. Originalni avtor unixa je bil sicer njegov prijatelj Ken Thompson, ki bi ga lahko prav tako označili za računalniškega polboga. Thompson je unix napisal v zbirniku (angl. assembler), a šele potem ko sta ga z Dennisom ponovno napisala v jeziku C, je unix postal zares prenosljivi operacijski sistem, ki je deloval na različnih računalniških sistemih. Ritchie je za svoja dela prejel enajst nagrad, od katerih naj v tem članku omenim le dve:

- 1983 prestižno Turingovo nagrado, ki jo vsako leto podeljuje Zveza za računske stroje (angl. Association for Computing Machinery – Turing award) in jo nekateri imenujejo kar Nobelova nagrada v računalništvu;
 - 1998 Nacionalno nagrado Združenih držav za tehnologijo (angl. U.S. National Medal of Technology), ki mu jo je izročil takratni predsednik ZDA, Bill Clinton.
- Zelo verjetno se bo vegovcem, ki



so se vpisali na našo šolo predvsem zaradi programiranja, na tem mestu prikradla misel: »C je itak že fosilna zadeva, če že moram kaj sprogramirati, raj uporabim Javo al pa C#, k sta ful bol u izi, za web sta itak bolj PHP al pa Python. Doma sm naložu windowse, k so mi kul, drgač mam pa itak najraj kakšn hud gejming, tko da ne vem, kva ma zdej ta Ritchie kle veze z mano ...«

Pa pojdimo po vrsti: čeprav se je o njegovih pomanjkljivostih šalil še sam Dennis, je jezik C eden izmed redkih jezikov, ki je preživel in se uporablja še danes, po več kot 40 letih od njegovega nastanka. Pa ne samo da se uporablja, po informacijah, ki jih je IEEE (Institute of Electrical and Electronics Engineers) objavil na svoji spletni strani

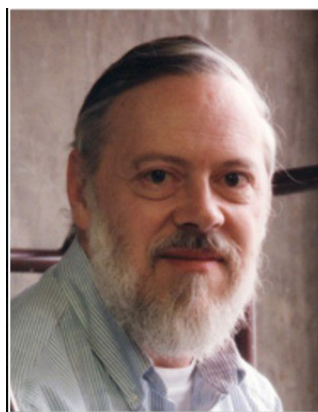
leta 2016, je ta še vedno najbolj popularen programski jezik na svetu. A bi bil preskromen, če bi se ustavil tukaj. Dejstvo namreč je, da je iz jezika C izpeljanih večina današnjih bolj modernih programskih jezikov: objektni C, C++, Java, C#, R, skriptna jezika Perl, PHP itd. Jeziki, kot sta Python, Ruby itd., pa so implementirani v jeziku C. Torej vsi, ki na različnih lestvicah popularnosti dosegajo prvih pet mest. Za navdušence računalniških iger pa naj samo napišem nekaj imen orodij za izdelavo iger (ang. engine), za katere mislim, da mi jih ni potrebno posebej predstavljati: Doom, Quake, Quake2, CryEngine, Crysis, IW engine (Call of duty), Unreal engine in še cela množica drugih so napisani v jeziku C ali pa C++. In da, tudi operacijski sistem (v nadaljevanju OS) Windows, na katerem je Bill Gates zgradil svoj imperij, je bil in je še danes (deloma) napisan v jeziku C.

Vegovci, ki jih zanima bolj strojna oprema, si bodo na tem mestu mogoče mislili: »Pa to programiranje je itak beda pa še ful je zatežen vse skupi. Jst mam raj robotke pa mikrokrmilnike pa Raspberry Pi, mobilna telefonija mi je pa itak zakon. Zaenkrat sm še na androidu, ampak mi je iPhone ful bol fensi, k ma iOS in si ga bom enkrat ubodu. Tale Dennis me s svojim unixom, k ga več noben ne uporablja, itak ne bo prepriču ...«

No, pa se zopet vprašajmo, na čem temelji vse skupaj. Operacijski sistem iOS, ki poganja tako Appleove telefone iPhone kot tablice iPad, temelji na sistemu OS X, ki ni nič drugega kot še ena izmed različic t. i. BSD Linux jedra. Večina konkurenčnih pame-

tnih telefonov ima naložen OS Android, katerega jedro si deli z OS Linux. Raspberry Pi uporablja OS Raspberian, ki bazira na t. i. Debian Linux OS. OS Linux je napisal 21-letni študent računalništva Linus Torvalds leta 1991 kot brezplačen OS za procesorje AT (80386/486), ki so jih takrat uporabljali osebni računalniki. Kljub temu da je bil napisan povsem na novo, močno spominja na unix in ga zato strokovnjaki radi poimenujejo kot OS »Unix like«. Iz slednjega seveda sledi logična posledica, da tako pametni telefoni kot imperij Steva Jobsa slonijo na Dennisovem delu. Glede robotike in mikrokrmilnikov naj samo napišem, da razvoj humanoidnih robotov in strojnega vida v industrijskem okolju prav tako pokriva programski jezik C ali C++ (npr. knjižnica za openCV, OCR – Optical Character Recognition itd.), mikrokrmilniki pa se (še vedno) pogosto programirajo v zbirniku ali pa v kakšni izmed inč jezika C (npr. mikroC).

Čeprav sem že veliko napisal o prispevku Dennisa Ritchieja računalniški vedi, mi dovolite, da na tem mestu navedem še citat enega izmed programerskih legend, ki trenutno dela za Google, g. Roba Pika, ki meni, da je Ritchiejevo delo osnova tudi za delovanje medmrežja: »Pravzaprav skoraj vse v medmrežju deluje na podlagi dvojega: unixa in C-ja. Brskalniki (npr. Chrome, Mozilla itd.) so napisani v jeziku C. Strežniški programi so večinoma zapisani v C-ju, če pa ne, so napisani v javi ali v C++, ki sta izpeljana iz jezika C, ali pa so napisani v jeziku Python ali Ruby, ki sta oba implementirana v jeziku C. Upam si trditi, da vso omrežno strojno opre-



UNIX is basically a simple operating system, but you have to be a genius to understand the simplicity.

— Dennis Ritchie —

AZQUOTES

mo krmilijo programi, napisani v jeziku C. Res je težko oceniti, koliko modernega informacijskega gospodarstva temelji na Dennisovem delu.« Sam na tem mestu še dodajam: tudi večina programske kode iskalnika Google je napisana v jeziku C++.

Zaradi vsega napisanega ga nekatere strokovnjaki po pomembnosti enačijo z Nikolo Teslo v elektrotehniki, zopet drugi menijo, da brez Dennisa ne bi bilo modernega računalništva in ne moderne programske opreme.

In sedaj vprašanje za milijon dolarjev: »Zakaj potem Dennis Ritchie ni poznan širši množici, če je njegov prispevek računalniški vedi res tako neizmerljivo velik?«

Lahko bi zapisali, da je genialnost tako Steva Jobsa, Billa Gatesa kot tudi Marka Zuckerberga predvsem v tem, da znajo ponuditi izdelke (tako strojne kot programske), ki so množicam všeč, imajo občutek za okus ljudi in nam znajo ponuditi tehnologijo, na katero se hitro navadimo, tehnologijo, ki nam je nekako pisana na kožo. In pri tem seveda ne moremo mimo dejstva, da so (ali pa so bili) zraven še izjemno nadarjeni poslovneži. Po drugi strani je Dennis Ritchie izumil tisto, kar je očem nevidno. Njegov jezik

in operacijski sistem se nahajata v samem jedru računalniških sistemov, ki se jih povprečen uporabnik ne zaveda, pa čeprav jih uporablja vsak dan. Stvari, ki jih jemljemo kot samoumevne, pa jih ne bi smeli. Ali kot je zapisal računalniški zgodovinar Paul E. Ceruzzi: »Dennis Ritchie ni bil opažen. Njegovo ime nikakor ni bilo poznano, a če bi imeli mikroskop in bi lahko pogledali v notranjost računalnika, bi videli njegovo delo prav povsod.«

Vzrok za njegovo nepoznanost med širšim občinstvom pa se skriva tudi v njegovem skromnem značaju. Dennis se kljub vsem priznanjem in nagradam, ki jih je prejel v svojem življenju, ni imel za nekaj več. Ko je opazil, da so ga nekateri njegovi oboževalci poimenovali polbog, je v hecu dejal: »Nisem in tudi nikoli nisem bil član polbožje stranke.« Dennis ni izumljal, da bi ustvaril zaslužek, vreden več milijard dolarjev, pač pa je to počel zato, ker je v tem res užival. Ustvarjal je v dobro nas vseh, z namenom, da se bomo vsaj na področju računalništva nekoč imeli bolje. Zato je prav, da smo s tem seznanjeni, da spoštujemo Dennisa Ritchieja in njegovo delo in ga cenimo vsaj tako kot prispevke ostalih računalniških polbogov.

Avtonomna vožnja

Matej Pevec, G 2. B

Lansko šolsko leto sem napisal obsežno raziskovalno nalogo s področja računalniškega vida in algoritmov za iskanje poti in pridobil kar veliko znanja, povezanega s tem. V današnjem času postajajo električna vozila, lahko rečemo, kar modna niša v avtomobilizmu in posledično s tem prihaja tudi avtonomna vožnja. Električne motorje je mnogo lažje upravljati z elektriko, saj jih ta poganja že v osnovi, v nasprotju s trenutnimi, ki se poganjajo s fosilnimi gorivi in katerih nadzor je mnogo težji.

Avtomatsko upravljanje vozila se, kljub temu da se sliši preprosto, izkaže, da je bolj kompleksno, kot bi si kdo predstavljal. Vsi avtomobili z avtomatsko vožnjo imajo vgrajeno zelo zmogljivo tehnologijo za procesiranje slik, ki jih pridobijo z več kamer, razporejenih po avtomobilu. Naloga teh procesnih enot je tako zaznavanje pešcev in avtomobilov kot tudi zaznavanje in prepoznavanje znakov in ceste. Te podatke potem procesor kombinira s podatki o lokaciji cest, ki jih avtomobil prenese s satelitov. S pomočjo posebnih namenskih satelitov za lociranje moči signala in lokacije satelitov procesor v avtomobilu preračuna svoje koordinate.

Z računalniškim vidom sem najprej iskal labirint in ovire. Na list papirja sem narisal okvir in pot. Ta okvir sem s pomočjo transformacije matrice poravnal, da sem dobil pravokotno projekcijo.

Drugi del avtomatizacije vožnje pa je iskanje optimalne poti med

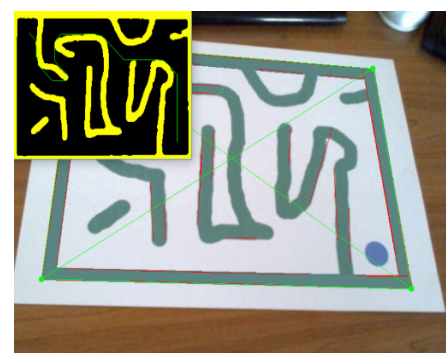
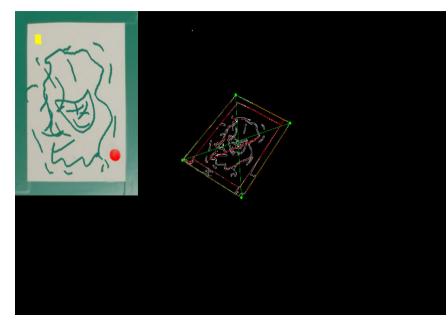
ovirami. Za to obstaja mnogo algoritmov, v katere sem se dodobra poglobil in poiskal najbolj optimalnega za opremo. Primerjal sem Dijkstrin algoritem z njegovim naslednikom A^* , nato pa sem iskal še izboljšave zanje. Na koncu sem dobil tako imenovani Jump Point Search algoritem ali na kratko JPS, ki je zelo učinkovit in se je tudi na manjših računalnikih, kot je Raspberry Pi, izkazal. V avtomobilih z vgrajeno avtonomno vožnjo je zelo verjetno vgrajen že kakšen bolj učinkovit algoritem. Za ovire sem vzel vnaprej transformirano sliko, izločil vso zeleno barvo in s pomočjo matematičnih in logičnih operacij izločil le uporabne ovire. Potem sem izločil še barvi, ki sta bili nastavljeni za cilj in začetek, in njuno povprečno lokacijo vzel za ti dve točki v iskalniku. Končni rezultat je bil transformiran in rešen labirint.

Avtomobili pa se ne ukvarjajo le s procesiranjem slik, temveč tudi videoposnetka. Postopek je zelo podoben, a mnogo bolj procesorsko zahteven, saj mora v realnem času izračunati enako količino podatkov kot prej. Na osebнем računalniku je to delovalo brezhibno, sem pa imel težave na vgradnem računalniku Raspberry Pi, saj je manjša zmogljivost omejila število procesiranih slik na eno na vsakih osem sekund, kar ne bi bilo uporabno v konkretnem avtomobilu.

Z raziskovalno nalogo sem želel raziskati in pokazati osnovni koncept delovanja avtonomne vožnje. Seveda je v ozadju še mnogo drugih logičnih in tudi matematičnih

operacij, ki avtomobilu dajejo podatke, kot sta oddaljenost od drugih avtomobilov, hitrost avtomobilov pred in za njim, a se v take podrobnosti ne bi preveč spuščal.

Avtomobilizem je zelo zanimivo področje, ki trenutno skokovito napreduje in bo v naslednjih letih prineslo še veliko novosti, o katerih so mnogi še pred kratkim samo sanjali ali pa jih uporabljali v znanstvenofantastičnih filmih kot nekaj, kar ogroža človeštvo ali pa je neizvedljivo. Danes pa vidimo, da je slika povsem drugačna in še kako realna.



Težave s pametnimi napravami

V petek, 21. oktobra 2016, nekaj ur ni bilo mogoče priti do Twitterja, Netflix, PayPal in drugih velikih spletnih strani in storitev, ker je prišlo do napada na DNS-strežnike. Ti delujejo kot imenik, v katerega vsaka elektronska naprava pogleda, ko želi dostopati do posamezne spletne strani ali storitve. Ko v brskalnik vpišemo ime spletne strani, računalnik vpraša DNS-strežnike, kje točno je ta spletna stran, in ti ga usmerijo nanjo. Brez DNS-strežnikov spletna strani preprosto ne obstajajo.

Ob takšnem napadu napadalci »vprežejo« nič hudega sluteče računalnike in jim ukažejo, naj posamezni strežnik zasujejo s podatki, in čakajo, da omahne pod ogromno količino zahtev.

Tokratni napad je bil nekaj posebnega. Izpeljali so ga z različnimi pametnimi napravami, zanje se uveljavlja

izraz »internet stvari«, ki so povezane v splet. To so pametni termostati, hladilniki, varnostne kamere in podobno. »Pamet« jim zagotavlja, jo vgrajeni mini računalniki, ki so redko ali pa sploh niso deležni varnostnih posodobitev. Ko napadalci odkrijejo šibko točko na tej pametni napravi, imajo kar naenkrat na voljo več v splet povezanih naprav. Povežejo jih v mrežo, ta pa nato napada njihove tarče.

In kako se boriti proti takim napadom? Proizvajalci bodo morali redno skrbeti za varnost svojih naprav. Prav tako je veliko odvisno od uporabnikov. Ti bodo morali ves čas posodabljeni računalnike in domače brezžične usmerjevalnike, enako pa bodo morali ravnati s pametnimi hladilniki, termostati, varnostnimi kamerami. Predvsem pa bo potreben resen premislek, kako proizvajalce pripraviti, da bodo nji-



hovi izdelki tudi z vidika povezanosti v splet varnejši.

Vir: Mladina, 28. 10. 2016 (Anže Tomič, S hladilniki nad internet).

Kruha in računalniških igr

Pred desetimi leti je Južna Koreja gostila svetovno prvenstvo v nogometu, zadnje čase pa so ti stadioni do vrha napolnjeni z oboževalci e-športov.

Po statistiki sodeč, se je število gledalcev med letoma 2014 in 2016 povečalo za 43 %, z 204 na 292 milijonov, do leta 2019 pa naj bi številka presegla 427 milijonov. Finančno gledano se je količina prihodkov povečala za 239 %, z 194 na 490 milijonov.

»V desetih letih bo enako NHL-u,« pravi Olof Kajbjer, nekaterim znan tudi kot Olofmaister. Istega mnenja

je tudi Carlos Rodriguez, ustanovitelj organizacije G2, ki pravi, da je kot Real Madrid e-športa.

Gledanost po svetu in prihodki kažejo na to, da bi videoigre lahko bile šport. Sam Mathews, ustanovitelj organizacije Fnatic, pravi, da so videoigre šport, pri katerem je računalnik zgolj vmesnik med dvema igralcema. Lahko so zelo strateški in od ljudi zahtevajo po 10 ur treninga na dan, da bi dosegli dobre rezultate.

Glede na to da so v Južni Koreji e-šport priznali kot olimpijsko disciplino, v ZDA igralci prejmejo enake vize kot vsi ostali športniki, CS:GO,

League of Legends in Dota 2 pa so se pojavili tudi na televiziji. Mislim, da bi jih morali priznati kot pravi šport po celem svetu.

Vir: Mladina, 5. 8. 2016 (Staš Zgonik, Do sedem ur na dan).



WinRAR

Stiskanje datotek

Jakob Pezdirc, R 1. C

(Mentor : Aleš Volčini, prof.)

Začetek

Prva različica programa je bila za javnost izdana 22. aprila 1995 in se posodablja vse do danes. Glavna, ki sta sodelovala pri izvedbi tega programa, sta bila Eugene Roshal in Alexander Roshal, in sicer Eugene kot razvijalec in Alexander kot glavni distributer. Danes je to najbolj popularen program za stiskanje datotek, ki ga uporablja več kot 500 milijonov uporabnikov.

Verzije

- Prva verzija se je pojavila že leta 1995 kot 16-bitna Windows 3.x aplikacija.

- Leta 2004 je izšla verzija (3.41), ki je dodala podporo za operacijski sistem Linux.

- Leta 2005 so z novo verzijo (3.50) začeli podpirati Windows XP(64x).

- Od leta 2008 je program začel podpirati ZIP-datoteke (verzija 3.80).

Podpora operacijskih sistemov

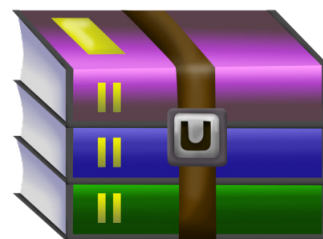
- WinRAR v4.11 je zadnja verzija, ki podpira Windows 2000.
- WinRAR v3.93 je zadnja verzija, ki podpira Windows 95,98,ME,NT in Windows mobile.

- RAR v3.93 je zadnja verzija, ki podpira MS-DOS in OS/2 na 32-bitnih procesorjih.

- RAR v2.50 je zadnja verzija, ki podpira MS-DOS in OS/2 na 16-bitnih procesorjih.

Logotip

Uradni logotip programa je sestavljen iz treh knjig različnih barv, ki so spete s pasom/trakom.



Kako deluje

WinRAR datoteke stisne tako, da večje »znake« spremeni v manjše oz. spremeni zapis tako, da zasede manj prostora. Za to uporablja »knjižnice«. Naj predstavim primer s citatom Johna F. Kennedyja:

»Ask not what your country can do for you - ask what you can do for your country.« Ta citat je sestavljen iz 61 črk, 16 presledkov, vezaja in pike. Če vsaka črka, presledek, vezaj in pika predstavljajo po eno

enoto spomina, dobimo 79 enot. Da program zmanjša število enot v tej povedi, išče vzorce in ponavljnja, ki bi jih lahko zapisal krajše, in za te krajšave ustvari knjižnico.

- Beseda "ask" se pojavi dvakrat
- Beseda "what" se pojavi dvakrat
- Beseda "your" se pojavi dvakrat
- Beseda "country" se pojavi dvakrat
- Beseda "can" se pojavi dvakrat
- Beseda "do" se pojavi dvakrat
- Beseda "for" se pojavi dvakrat
- Beseda "you" se pojavi dvakrat

Knjižnica:

1. ask
2. what
3. your
4. country
5. can
6. do
7. for
8. You

Naša poved sedaj, ko smo uporabili knjižnico, izgleda takole: »I not 2 3 4 5 6 7 8 -- 1 2 8 5 6 7 3 4« Število enot smo zmanjšali z 79 na 37,

endar moramo upoštevati, da je potrebno shraniti tudi knjižnico, ki pa zasede še 37 enot, skupno 74. Sedaj smo »datoteko« zmanjšali

za 5 enot, kar ni zelo veliko, vendar če bi bilo besedilo daljše, bi ga lahko procentualno zmanjšali še za veliko več. To je metoda iskanja

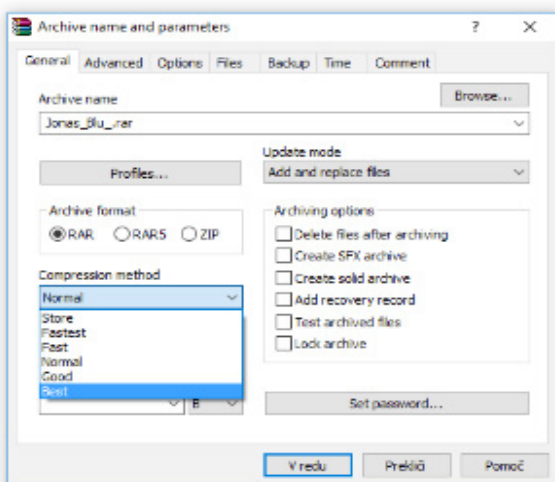
ponavljana, program pa išče tudi vzorce, na primer: »Stavki v slovenščini se tipično začnejo z vprašalnico npr. (...)« Tukaj program vidi,

da za tremi o-ji stoji presledek, zato se na primer odloči, da bo povsod, kjer o stoji pred presledkom, napisal samo en znak (eno enoto),

npr. 1. Za to bi v knjižnico napisal (1 = o). Knjižnica je kasneje pomembna zato, da program razširi dokument nazaj v prvotno stanje.

Uporaba

Stiskanje



Metode

Ime	Datum spremembe	Vrsta	Velikost
Jonas_Blu_	10/23/2016 10:24 ...	BSplayer file	9,165 KB
Jonas_Blu_	10/24/2016 9:41 PM	WinRAR archive	8,597 KB

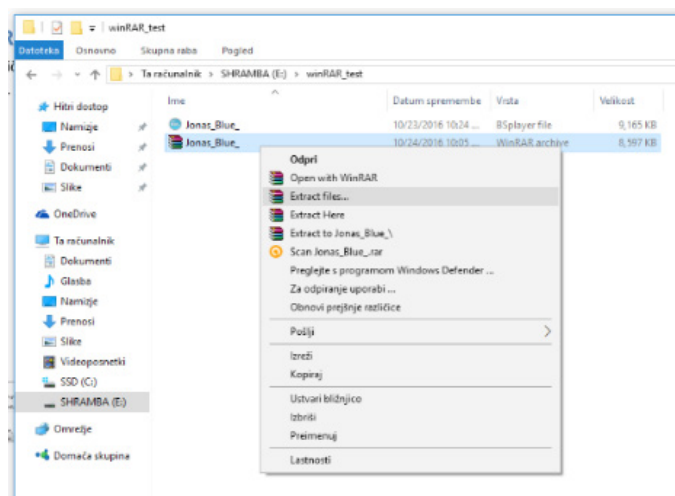
Razlika v velikosti

Če želimo neko datoteko na našem računalniku stisniti oz. spremeniti tako, da bo zasedla manj prostora, kliknemo na datoteko z desnim klikom, da se nam odpre okno,

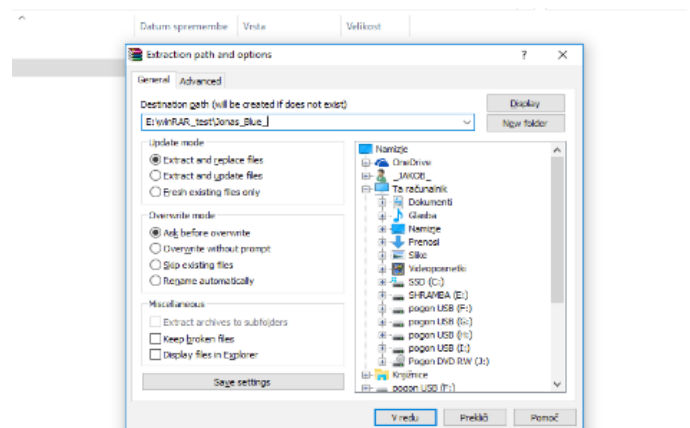
ki nam ponudi, da stisnemo datoteko. Ko izberemo to možnost, se nam odpre okno še z več možnostmi. Stisnjenost novonastale rar-datoteke je odvisna od vrste

(img, mp3, docx, ...) in velikosti datoteke, ki jo hočemo stisniti, in tega, kakšno metodo izberemo. Višja metoda, daljši čas stiskanja in bolj stisnjena rar-datoteka.

Razširjanje



Razširi sem

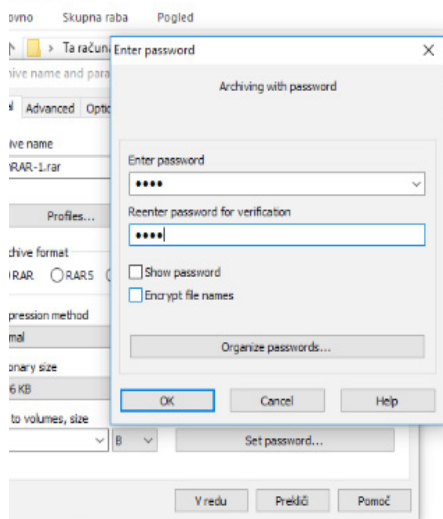


Razširi v določeno mapo

Razpakira različne vrste datotek (RAR, ZIP, CAB, ARJ, LZH, ACE, TAR, GZip, UUE, ISO, BZIP2, Z in 7-Zip)

Zaklepanje dokumentov

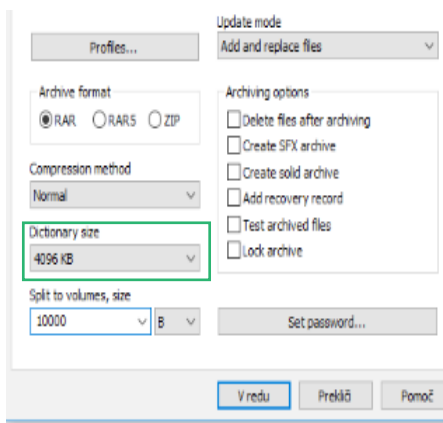
Če nekomu pošljemo bolj zaupne datoteke, lahko dokumente s pomočjo tega programa zavarujemo z geslom, kot kaže slika.



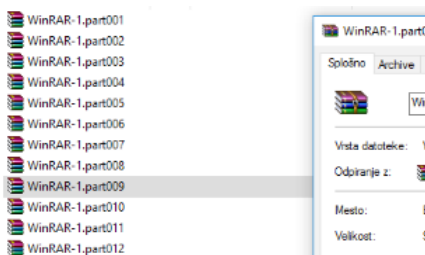
Nastavljanje gesla

Deljenje večjih datotek na več manjših

Če imamo shranjeno večjo datoteko in jo želimo poslati nekomu prek internetne povezave, kjer je omejena velikost datoteke (npr. gmail, dropbox, ...), jo z winRAR lahko razdelimo na več delov in jo nato pošljemo prejemniku po kosih. Ko prejemnik dobi vse dele, jih lahko razširi in dobi željeno datoteko.



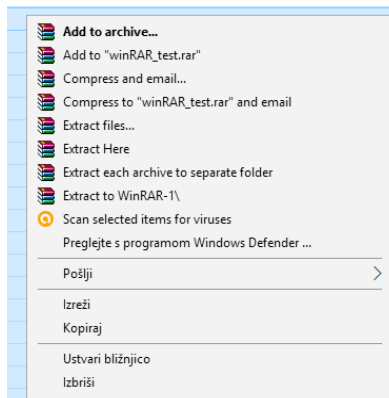
Deljenje datoteke na več delov



Nastanek pri delitvi datoteke

V tem prostoru (prostor, ki je označen na sliki) izberemo, kako velike razdelke želimo dobiti (prilagodimo glede na to, kako in kje bomo te datoteke pošiljali, shranjevali, ...). Velikost vpišemo v bajtih, kilobajtih, megabajtih ali gigabajtih.

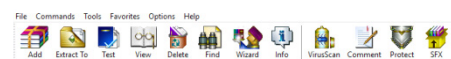
Nastane več datotek z izbrano velikostjo. Če potem vse rar-datoteke, ki smo jih dobili, označimo in pritisnemo desni klik, dobimo dodatne možnosti:



Možnosti z razdeljenimi rar-datotekami

- Datoteko lahko razširimo ali stisnemo kot pri ostalih rar-datotekah.
- Iz vseh rar-datotek lahko skupaj naredimo eno.
- Vsako rar-datoteko lahko razširimo v svojo mapo.
- Datoteke lahko razširimo in nato takoj pošljemo po e-pošti, vendar za to rabimo Outlook račun.

Več možnosti



Če odpremo datoteko s programom, dobimo še dodatne možnosti:

TEST (test): testiramo, če datoteka deluje pravilno

VIEW (vpogled): pogledamo, kaj je v mapi, ki jo imamo označeno (enako kot levi dvoklik)

DELETE (izbriši): brisanje datotek

FIND (najdi): vpišemo vrsto datoteke, njeno ime in podatek, kje naj to išče (C-disk, D-disk, ...); nato jo program najde

INFO (informacije): izpišejo se podatki o izbrani datoteki

VIRUS SCAN (pregled za virusi): pregleda datoteko, če vsebuje viruse

COMMENT (komentar): datoteko lahko komentiramo tako, da ko jo nekomu pošljemo, bo on to videl

PROTECT (zaščita): zaščiti datoteko

WinRAR ima možnost zastonj preizkusiti program (trial) za 40 dni, in čeprav program normalno deluje, tudi po izteku 40-ih dni z izjemo, da ti program ves čas ponuja plačljivo verzijo. Če program želiš kupiti, ga lahko kupiš za približno 40 €. Izjema je Kitajska, ki ima od leta 2015 zastonj celotno verzijo programa.

Konkurenca

- PowerArchiver
- WinZip
- 7-Zip

To so še drugi (jih je še veliko več) programi za stiskanje/razširjanje datotek, ki delujejo na zelo podoben način. Med njimi najbolj znan je 7-Zip.

Zaključek

Z WinRAR-om se srečujem skoraj vsak dan, ko iz partisa ali drugih spletnih strani povlečem kakšen program, film ali video igro. Zelo sem vesel, da obstaja, ker imam doma slabo internetno povezavo in mi zato WinRAR kar precej skrajša čakanje prenosa datotek (predvsem velikih, kot so igre). Nikoli nisem uporabljal drugega

programa za stiskanje/razširjanje datotek, ker sem bil z njim vedno zadovoljen.

Viri:

[https://en.wikipedia.org/wiki/](https://en.wikipedia.org/wiki/WinRAR)

WinRAR (18. 10. 2016)

[https://sl.wikipedia.org/wiki/](https://sl.wikipedia.org/wiki/Datote%C4%8Dni_format_RAR)

Datote%C4%8Dni_format_RAR

(8. 3. 2013)

<https://www.quora.com/How-exactly-does-compression-software-like-winrar-and-7-zip-work> (22. 11. 2012)

<http://www.technorms.com/14632/winrar-open-zip-rar-compressed-archives> (2016)

<http://computer.howstuffworks.com/file-compression3.htm> (2016)

<http://www.win-rar.com/features.html?&L=0> (2016)

Eksperimentiramo s ploščo Arduino

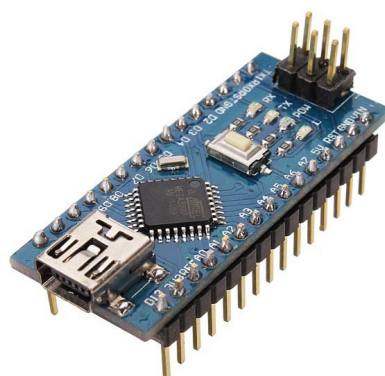
Andrej Sušnik, R 3. A

Arduino je skupno ime za družino mikroprocesorskih razvojnih orodij, ki so namenjena tako tistim, ki se z računalniško mikroelektroniko šele spoznavajo, kot tudi tistim, ki jo poznajo dobro in želijo z njo čim hitreje izpeljati samo neko idejo ali kar cel projekt.

V 10-ih letih obstoja je Arduino osvojil učilnice po vsem svetu, entuziastom pa ponudil poceni in hitro razvojno orodje. Na svet elektronike in računalništva je naredil tako močan vtis, da ga ne morejo več ignorirati niti tako velika podjetja, kot sta npr. Intel in Microsoft. Združljive in popolnoma funkcionalne izdelke za nizko ceno ponujajo tudi najrazličnejši proizvajalci Daljnega vzhoda.

Arduino Nano

Plošče Arduino prihajajo v več izvedenkah, najpogostejši sta izvedenki Uno in Nano. Za naš izdelek bomo uporabili izvedenko Nano, ki je malo manjša in se uporablja za izdelke, pri katerih varčujemo s prostorom.



Na plošči 13 priključkov služi vhodnim ali izhodnim napravam, kot so npr. stikala in LED-diode, 6 priključkov pa omogoča priklop analognih vhodnih naprav. Tu je še posebej USB-priključek za napajanje in komunikacijo z osebnim računalnikom. Kadar pa zraven ne potrebujemo računalnika, ga lahko napajamo z navadnim USB-polnilcem. Potrebujemo še okolje za programiranje.

Priprave

Naše osnovne potrebščine so:

- plošča Arduino Uno
- USB A-Mini B-kabel
- program Arduino (<https://www.arduino.cc/en/Main/Software>)
- kratke žice različnih barv

Potrebujemo samo še elektronske komponente za eksperimentiranje.

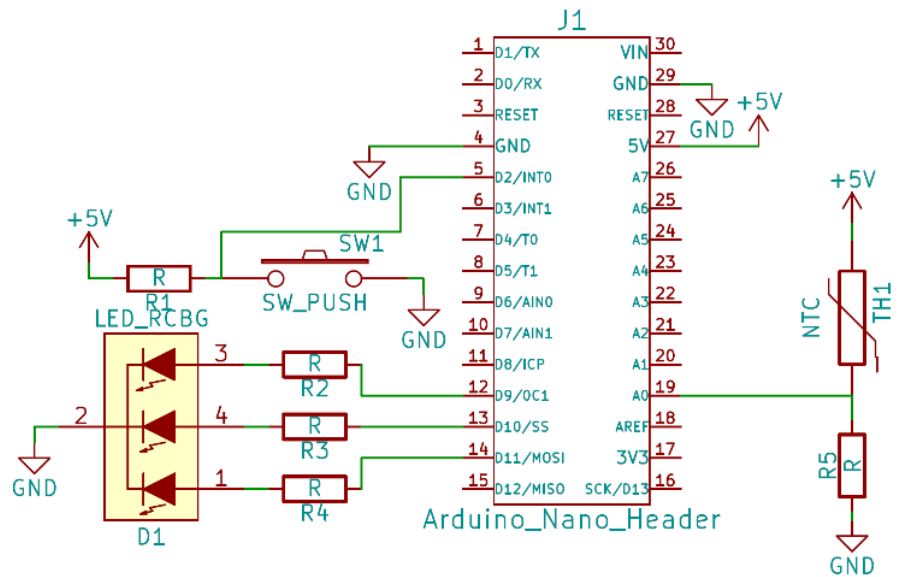
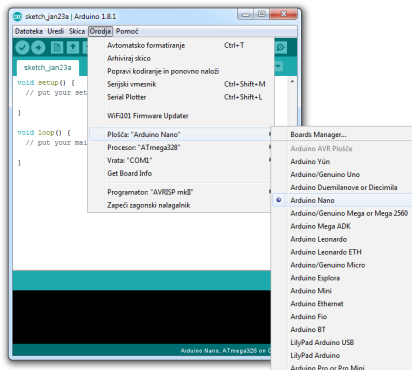
Arduino in testiranje povezave

Urejevalnik programa se preprosto imenuje Arduino. Je običajen urejevalnik za programski jezik C, v katerem napišemo program in ga pošljemo na vezje.

1. Arduino priključimo na računalnikov USB.
2. Odpremo urejevalnik. V njem nas že čaka preprost program.
3. Preverimo, da imamo pod zavihkom orodja pri plošči izbran arduino nano, kot kaže slika.
4. Pred pošiljanjem se prepričamo, da imamo izbrana prava COM-vrata. Po navadi so to vrata COM2 ali COM3 in so izpisana v spodnjem desnem kotu urejevalnika.
5. Pritisnemo gumb Naloži (na sliki drugi gumb v zgornjem levem kotu).
6. V oknu se izpiše obvestilo, ali je bilo nalaganje uspešno ali ne. Če se arduino ne odziva, preverimo, ali je priključen in ali so izbrana prava vrata.

Potreben material

Naše multifunkcijsko vezje bo imelo štiri načine delovanja, med katerimi bomo izbirali s pritiskom na tipko.



V prvem načinu bo deloval kot okrasek in bo naključno menjal barve, v drugem načinu bo menjal barve tako kot semafor, v tretjem se bo odzival kot termometer za hladno ali vroče (menjal bo barve med rdečo in modro) in v četrtem bo sporočal temperaturo – številko dolгих svetlobni signalov bodo desetice, kratkih pa enice.

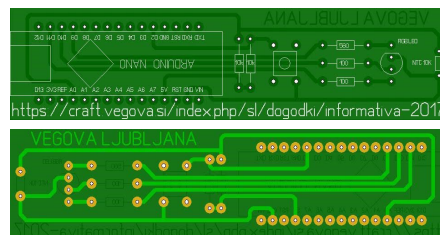
Za vezje potrebujemo še:

- 3-barvno diodo (RGB)
- upornik, odvisen od temperature (NTC-termistor)
- 2 upornika 10 k Ω
- 2 upornika 100 Ω
- upornik 560 Ω
- tipko

Vezava

Za naše multifunkcijsko vezje elemente vezemo, kot kaže slika zgoraj. Dioda (v rumenem pravokotniku) je preko treh upornikov (mali pravokotniki) vezana na nogice arduina, katoda (najdaljša nožica) diode pa na GND (ozemljitev). Termistor (označen s TH1) imamo prek upora vezan na t. i. analogni vhod A0. Tipka je označena s SW_PUSH in je po

eni strani vezana na upor in nogico D2 arduina, po drugi pa na GND. Še žice: v našem primeru bodo nadomestile bakrene povezave na ploščici. Taki ploščici pravimo tiskano vezje in jo vidite na slikah spodaj. Prva slika je slika vezja z zgornje strani, kjer imamo narisano, kam sodijo kateri od elementov. Druga slika pa kaže povezave na spodnji strani.



Program Nastavitve

Program je zelo preprost. V proceduri setup povemo, da bomo

diodo prižigali prek nogic 9, 10 in 11 in da bodo te nožice imele vlogo izhodne naprave.

Te nožice smo izbrali z razlogom: podpirajo t. i. pulzno širinsko modulacijo – PWM (Pulse-width modulation).

Razloženo na hitro: Nogice, ki ne podpirajo PWM-ja, so lahko samo v dveh stanjih: 0 in 1. Če na njih merimo napetost, to pomeni 0 V ali 5 V. PWM-pini lahko menjajo 0 in 1 v zelenem razmerju in danem času (recimo stotinka sekunde). Izbiro navedemo v ukazu AnalogWrite (glej program): če je številka 255, je na nogici ves čas napetost 5 V, če je 127, potem je 5 V skoraj točno polovico časa, drugo polovico je 0 V. Če uporabimo število 0, je na nogici ves čas 0 V.

```
void setup(){
  Serial.begin(9600);
  pinMode(redPin, OUTPUT);
  pinMode(greenPin, OUTPUT);
  pinMode(bluePin, OUTPUT);
  pinMode(button, INPUT_PULLUP);
}
```


Ta lastnost je zelo uporabna. S takim načinom prižiganja LED-diode dosežemo, da se nam zdi, da LED gori na polno, na polovico, da je ugasnjena ali da je nekje vmes, pri tem pa vezju ni potrebno dodati ničesar.

V našem primeru bomo s pomočjo PWM-ja generirali različne barve – izberemo lahko, kako močno sveti katera od treh barv. V RGB LED-diodi se namreč skrivajo kar tri diode: rdeča, zelena in modra.

```
analogWrite(redPin, 51);
```

V zgornjem primeru bo rdeča dioda prižgana v razmerju 51 : 255 oziroma približno četrtno časa. To pomeni, da bo gorela s približno četrtno svoje polne svetlosti.

Naslednji del je procedura loop, ki jo arduino ponavlja, vse dokler ga ne izklopimo. V njej kličemo funkcijo main1(), ki izbira med načini delovanja.

```
void loop()
{
    main1();
}
```

Da bi si pogledali ves program, imamo premalo prostora. Pogledajmo pa si en njegov del.

Vizualni termometer

Termometer, s katerega odčitamo odtenke, ne pa temperature. Zakaj pa ne?

Imamo termistor – element, ki prepušča več ali manj električ-

nega toka glede na to, kolikšna je temperatura na njegovem površju. Tok potuje skozi drugi upornik, na katerem se zaradi tega pojavi napetost. To napetost pa merimo na analognem vhodu.

V programu napetost merimo s pomočjo funkcije analogRead. Ta nam vrne neko številko od 0 do 1023 – večja, kot je številka, višja je napetost.

Če hočemo videti sprotne rezultate meritev, s funkcijo SerialWrite pošljemo podatek osebemu računalniku, kjer lahko vidimo, kako se števila spreminjajo glede na temperaturo termistorja.

Sedaj se moramo samo še odločiti, pri kateri vrednosti bo dioda svetila modro, pri kateri pa rdeče.

Program

```
/*Funkcije toplota*/
/*Glede na temperaturo prilagodi osvetlitev
20 stopinj..... analogna vrednost 20
35 stopinj .....analogna vrednost 255
*/
void toplota()
{
    int value = analogRead(termistor); //V spremenljivko value zapišemo
    vr ednost ki smo jo dobili iz termistorja
    double temp=Thermistor(value); //V spremenljivko temp zapišemo vrednost ki
    jo poda funkcija Thermistor ki kot vhodni parameter dobi spremenljivko value
    Serial.println(temp);
    int rdeca = map(temp, 20, 35, 20, 255);
    int modra = (255-rdeca);
    if(temp>=35) //Če je temperatura večja ali enaka 35 stopinj
    {
        analogWrite(redPin, 255);
        analogWrite(bluePin, 0);
        delay(500);
        analogWrite(redPin, 0);
        analogWrite(bluePin, 0);
    }
    else //Če je temperatura manjša od 35 stopinj
    {
        analogWrite(redPin, rdeca);
        analogWrite(bluePin, modra);
    }
    delay(100);
}
```

Arduinova sporočila si lahko ogledujemo v oknu Serijski vmesnik, ki ga dobimo v meniju Orodja. Paziti moramo le, da imamo na njem nastavljeno enako hitrost kot v programu (9600 baud).

V nadaljevanju se glede na spremenljivko temp odločamo, kako močno bosta svetili rdeča in modra dioda.

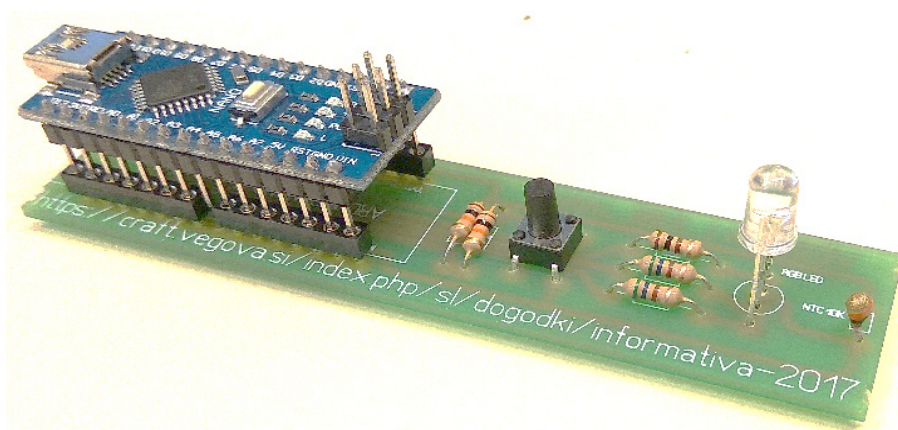
Izboljšave

Obstoječe vezje lahko izboljšamo:

- lahko dodamo prelivanje med več barvami,
- naredimo lightshow,
- lučka posveti, ko temperatura naraste nad določeno mejo ali pade pod njo ...

Program in vse ostalo boste našli tudi na spletni strani:

<https://craft.vegova.si/index.php/sl/dogodki/informativa-2017>



Končni izdelek

Kaj vse še preizkušamo na arduinu?

V zgornjih primerih smo videli, kako priklopimo in upravljamo LED-diode in spremljamo stanje senzorja temperature. S preprostimi elektronskimi elementi pa lahko naredimo še mnogo poskusov. Arduinov urejevalnik ponuja ogromno primerov. Med drugim se

lahko sami (ali pa tudi na naši šoli) naučite, kako se priključijo, delujejo in kako se uporabljajo: fotouporniki, stikala in tipke, potenciometri, enobarvni in večbarvni LED-zaslone, termistorji (kot senzorji temperature), ultrazvočni senzorji (za merjenje razdalje), akumulatorji in polnilci zanje, GSM, bluetooth in WiFi-moduli in še in še ...

Cisco akademija

David Grabnar, R 3. B

Vegova v sodelovanju s podjetjem Cisco, ki je vodilno na področju telekomunikacij, organizira tečaj Akademije Cisco, ki je namenjen tako dijakom kot tudi študentom. Na tečajih udeleženci pridobijo teoretično znanje, ki ga nato z laboratorijskimi vajami tudi utrdijo, saj delajo na resničnih na-

pravah, s katerimi bi se srečali na delu.

Za tečaj CCNA ni potrebno predznanje, zato je odlično za tiste, ki šele odkrivajo svet omrežij, saj celovito pokriva osnovna in tudi napredna področja omrežij. Udeleženci lahko tečaj opravijo

v 2 letih, po končanem tečaju pa so usposobljeni za vzdrževanje manjših omrežij.

Vsak modul traja pol šolskega leta, sestavljen pa je iz več poglavij. Najprej se predela teoretični del poglavja, nato pa mu sledi še laboratorijska vaja, s katero se lahko teorijo prenese v prakso na pravih napravah.

Cisco certifikat

Certifikati so hierarhično naravnani. Na Vegovi lahko opravimo tečaj na področju CCN Associate, ki pa je odlična podlaga za nadaljnje izobraževanje in pridob-

Tečaj CCNA (Cisco Certified Network Associate)

- CCNA 1: Osnovni koncepti omrežij
- CCNA 2: Osnove usmerjevalnih protokolov
- CCNA 3: Protokoli na nivoju stikal in brezžičnih omrežij
- CCNA 4: Delovanje prostranih omrežij

bitev certifikatov, ki so še višje v hierarhiji.

Trenutno dokončujem 3. modul tečaja, ki se je začel s šolskim letom 2016/2017. Za tečaj sem se odločil v 2. letniku, ker so me zanimala omrežja in sem hotel izvedeti še kaj več, kot smo se naučili pri rednem pouku. Čeprav sem bolj zainteresiran za programiranje kot za omrežja, sem s tečajem zelo

zadovoljen, saj so ure zanimive, včasih tudi zahtevne, vendar nikoli nemogoče, saj nam inštruktor vedno razloži snov na konkretnih primerih. Velik del k uspehu prispevajo tudi laboratorijske vaje, kjer se seznanimo s težavami, ki jih ne bi pričakovali, če bi naše znanje temeljilo samo na teoriji. Končni izpit je sestavljen iz teoretičnega dela, praktičnega dela in domačih nalog, ki h končnemu se-

števku prispevajo le majhen delež, a je ta včasih lahko tudi odločilen.

Tečaj priporočam vsakemu, ki ga zanimajo omrežja, ne glede na predznanje. Tečaj zaradi obsežnosti zahteva nekaj dela tudi doma, ki pa je minimalno, če je narejeno sproti. Prepričan sem, da mi bo ta tečaj prišel prav, če ne drugače, pa s tem, da obvladam celotno snov omrežij 3. letnika, še preden jo obravnavamo pri rednem pouku.

Hanse – evolucija sodelovanja

Urh Vovk in Rok Strah, R 3. C

(Mentorica: Melita Kompolšek, prof.)











Hanse je bila trgovska zveza v poznem srednjem veku, v katero je bilo vključeno veliko mest današnje Nemčije, Poljske, Švedske, Nizozemske, Estonije, Latvije in Rusije. Njen center je bil Lübeck. Ta zveza je delovala na področju Baltika in Severnega morja, kjer je imela tudi monopol. Mesta so med seboj sodelovala, saj so tako vsi izboljšali svoj položaj.

Podobno sodelovanje predstavlja tudi igra *Zapornikova dilema*. To je igra, v kateri sta igralca obsojena zločinca, ki ju zaslišuje policija. Če oba zločinca priznata, dobita vsak po leto zavora. Če eden izmed igralcev prizna, medtem ko drugi ne, potem dobi tisti, ki je priznal, štiri leta zavora, tisti, ki ni priznal, pa je izpuščen. Pod pogojem, da nihče od igralcev ne prizna zločina, dobita vsak po dve leti zavora. Zato sledi, da je za oba igralca najboljša izbira sodelovanje; enako je veljalo tudi v trgovski zvezi Hanse. Seveda pa so takšne dileme, sodelovati ali ne sodelovati, znane tudi v vsakdanjem življenju (npr. doping

na športnih tekmovanjih, goljufanje pri igrah itd.).

Na podlagi opisanega je nastal na šoli krožek, pri katerem ustvarjamo simulacije trgovanja v zvezi Hanse z načeli zapornikove dileme. Trenutno izdelujemo tri igre, ki simulirajo trgovanje. Prva je namenjena več igralcem; zmagovalec je tisti, ki najdlje zdrži, ne da bi izgubil svoje bogastvo. Druga igra deluje na računalniku Raspberry Pi, s katerim so povezani gumbi, ki predstavljajo sodelovanje oz. nesodelovanje; zmagovalec je tisti, ki po določenih rundah pridobi večje bogastvo. Tretja igra je prenosna verzija druge igre, ki bo delovala na mikrokrmilniku Arduino Uno in bo prav tako imela gumbe, ki bodo predstavljali sodelovanje oz. nesodelovanje.

Krožek je nastal kot produkt sodelovanja partnerskih šol Berufsskolleg Niederberg (Velbert, Nemčija), OSEKK (Oulu, Finska) in IES ANDRES VANDELVIRA (Albacete, Španija) v okviru projekta *Evolu-*

ZAPORNIKOVA DILEMA		Izda	Molči	
				
Izda	 2 leti	 2 leti	 0 let	 4 leta
Molči	 4 leta	 0 let	 1 leto	 1 leto

tion of cooperation – computer simulations of trade and interaction in se bo izvajal v štirih fazah. Trenutno sodelujemo v prvem delu, v katerem izdelujemo simulacije srednjeveškega trgovanja. V drugi fazi bomo svoje igre optimizirali in jih distribuirali po spletu. Nato bomo v tretji fazi pridobljeno znanje in veščine uporabili pri šolskih predmetih in v vsakdanjem življenju. Opisane faze bo vsaka partnerska šola izvajala individualno, v četrtem delu pa bomo vsi udeleženci pripravili skupno dokumentacijo, da bodo lahko igre preizkusile tudi druge šole po svetu.

Raspberry Pi 3B in operacijski sistem na zunanjem disku ali internetnem omrežju

Aleš Volčini, prof.

V računalniku Raspberry Pi služi kartica SD kot preprost zunanji pomnilnik, vendar pri njeni uporabi zelo pogosto prihaja do okvare datotečnega sistema. Najpogosteje se to zgodi takrat, ko nenadoma zmanjka elektrike, ko računalnik pri eksperimentiranju zamrzne ali ko ga izklopimo tako, da mu nenadoma vzamemo elektriko. Pred časom so v programski kodi našli še hrošča, ki je povzročal okvare tudi ob pravilni uporabi.

Vsaka nepravilna zaustavitev za seboj potegne preverjanje ali popravljanje datotečnega sistema ali kar odpoved sistema. Še posebej je to nadležno, kadar eksperimentiramo z več računalniki,

denimo povezanimi v gručo ali pri delu v razredu. Včasih je lažje kartico kar formatirati in operacijski sistem naložiti na novo, kot pa se ukvarjati s popravili in ugotavljanjem, katere datoteke so okvarjene.

Računalniška skupnost si je že dolgo želela, da bi se lahko operacijski sistem nalagal ne samo s kartice SD, ampak da bi se ga dalo naložiti tudi s kakšne druge naprave, npr. zunanjega diska ali celo računalniškega omrežja, in da bi se ga dalo uporabljati tudi brez lastnega zunanjega pomnilnika. Pojavile so se tudi avtomatizirane rešitve (eno najdete na strani Adafruit), ki so omogočale osnovni zagon OS s kartice SD, ki

mu je sledil preklop na zunanjo USB-napravo.

Raspberry Pi 3 ima na voljo 32 kB flash pomnilnika za zagonski blok, kar je za današnje razmere smešno majhna količina pomnilnika. Običajno je notri program, ki začne z nalaganjem operacijskega sistema s kartice SD. V teh 32 kB je bilo potrebno spraviti program, ki lahko nalaga operacijski sistem s kartice, masovne pomnilniške naprave ali internega omrežja. Lansko poletje je programer Gordon Hollingworth objavil, da je pripravil zagonski program, ki omogoča nalaganje sistema z USB--naprave ali internega omrežja tudi brez uporabe SD-kartice.

Pripravimo Raspberry Pi 3

Da bi lahko uporabljali Raspberry Pi 3 z zunanjim pomnilnikom ali datotečnim sistemom na omrežju, potrebujemo:

- Raspberry Pi 3B
- SD-kartico
- strežniški računalnik z OS Ubuntu 16.04 LTS in programe:
 - DHCP ali DHCP proxy strežnik
 - TFTP-strežnik
 - NFS

Potrebno bo tudi nekaj znanja ukazne vrstice v linuxu in vsaj urejevalnika nano.

Raspberry Pi in zagonski blok

Čisto na začetku moramo zagonski blok posodobiti na neko

različico, ki omogoča nalaganje OS iz alternativnih virov. Zato potrebujemo SD-kartico z že na-

loženim operacijskim sistemom in že instaliranim programom `rpi-update`:

```
sudo apt-get install rpi-update
sudo rpi-update
```

Če želimo delati z eksperimentalno različico zagonskega bloka, uporabimo namesto zadnjega ukaza ukaz (previdno, to lahko za sabo potegne probleme):

```
sudo BRANCH=next rpi-update
```

Naslednji korak je nepovraten: ko je nastavek vpisana, poti nazaj ni.

Sedaj moramo v OTP (one-time-programmable, enkratno programirljiv pomnilnik) vpisati še

eno enko. Ta bo zagonskemu programu povedala, naj v primeru, ko ni kartice SD (ali pa je prazna), poskuša operacijski sistem naložiti najprej z zunanje USB-naprave (ključek, trdi disk), nato pa še z internetnega omrežja. To storimo

tako, da na konec datoteke `config.txt` dodamo vrstico `program_usb_boot_mode=1`.

Najhitreje to storimo z ukazom `echo` in preusmeritvijo izpisa na konec datoteke:

```
sudo echo program_usb_boot_mode=1 >> /boot/config.txt
```

Sedaj ponovno zaženemo RPi s `sudo reboot`. Zaželeno je, da po ponovnem zagonu to vrstico z nekim urejevalnikom odstranimo, še posebej če bomo kartico uporabljali še v kakem drugem RPi-ju.

Sistem je zdaj pripravljen. Pri zagonu zdaj poizkuša naložiti operacijski sistem v naslednjem vrstnem redu:

- *kartica SD (tako kot prej)*
- *zunanja USB-pomnilniška enota – ključek ali trdi disk*
- *internetno oz. intranetno omrežje*

S tem še niso rešeni vsi problemi; zagon s trdih diskov praviloma (še) ne dela (lahko da je potrebna še dodatna nastavek za čakanje, `program_usb_timeout=1`), za zagon z internetnega omrežja pa je potrebno imeti preklopnik (switch), ki se od-

zove takoj in ne meče prvih omrežnih paketkov stran.

Dobra stran je, da nalaganje z omrežja deluje vedno, če vstavimo kartico, formatirano na FAT32, na njej pa samo datoteko `bootcode.bin`, ki smo jo dobili ob posodobitvi z `rpi-update`. Tako kartico lahko uporabimo tudi na RPi-serij od B+ dalje. Slaba stran je, da težko ločimo med več RPi-ji, ki nalagajo sistem z omrežja, se pa nekako da.

Nalaganje sistema z USB-pomnilnika

Če želimo sedaj namesto SD-kartice uporabljati USB-ključek, najprej nanj skopiramo linux distribucijo na isti način,

kot bi jo nalagali na SD-kartico. Najlažje to storimo s programom `Win32Diskimager` na OS Windows, na linuxu pa s pro-

grami, kot sta `Etcher` ali `ImageWriter`. Več o tem na strani http://elinux.org/RPi_Easy_SD_Card_Setup.

Nato je potrebno storiti še štiri stvari:

- V mapi `boot` poiskati datoteko `cmdline.txt` in besedilo `root=/dev/mmcblk0p2` nadomestiti z `root=/dev/sda2`. Če boste uporabljali več ključev, jih je potrebno priklopiti na USB na prava mesta, da naprava `/dev/sda` pade na pravo mesto. Zanesljivejši način je uporaba UUID-diska.
- Če imamo v tej datoteki še `init=/usr/lib/raspi-config/init_resize.sh`, ga zberišemo.
- Zdaj poskusimo naložiti sistem, a pride do napake, ker v datoteki `/etc/fstab` priklopi še vedno kažejo na spominsko kartico. Malo počakamo, končno sistem javi napako in nam da ukazno vrstico. Zdaj v datoteki `fstab` popravimo še priklope, tako da `mmcblk0p1` popravimo v `sda1` in `mmcblk0p2` v `sda2`. Priklop /proc pustimo na miru.

```
sudo nano /etc/fstab
```

Prej:

```
/dev/mmcblk0p1 /boot vfat defaults 0 2
/dev/mmcblk0p2 / vfat defaults 0 2
```

Potem:

```
/dev/sda1 /boot vfat defaults 0 2
/dev/sda2 / vfat defaults 0 2
```

- Končno razširimo drugo particijo (/) do konca prostega prostora:

```
sudo touch /forcefsck
sudo parted /dev/sda resizepart 2 100%
sudo reboot
sudo resize2fs /dev/sda2
```

Če je šlo vse po načrtu, RPi deluje tudi brez SD-kartice na zunanjem pomnilniku.

Nalaganje sistema z internetnega oz. intranetnega omrežja

Nalaganje OS z omrežja je še zelo v eksperimentalni fazi, zato postopek ne bo delal s katerokoli različico zagonskega bloka, tudi težav bo obilo. Ta del je za ljudi s precejšnjo mero znanja in dobršno mero potrpežljivosti, saj gre lahko marsikaj narobe.

Pripraviti moramo še strežniške programe storitev DHCP, TFTP in NFS. Privzeli bomo, da imamo OS Linux Ubuntu 16.04 LTS. Izberemo seveda lahko tudi katero drugo distribucijo, razlike pa bodo pomenile tudi kakšno neprijetnost več.

DHCP-namestniški strežnik

Ko priključimo napravo v interneto ali intranetno omrežje, mora od nekje pridobiti svoj internetni naslov. Vedno ji lahko dodelimo in nastavimo statični naslov, vendar to zahteva nekaj znanja in menda

ni primerno za vsakogar.

Drugi način je s pomočjo strežnika DHCP. Njegova naloga je, da napravi ob priklopu dodeli

IP-naslov in ji posreduje podatke o omrežju, na katerega je priključena. Ti podatki med drugim vsebujejo tudi naslov strežnika TFTP, preko katerega je mogoče dobiti datoteke, potrebne za zagon in delovanje operacijskega sistema.

Tu se pojavi še en problem: DHCP-strežniki so zdaj večinoma vgrajeni in vključeni na mrežnih usmerjenikih in ne podpirajo posebnih nastavitev, povrhu pa do njih nimamo vedno dostopa.

Sedaj nastopi DHCP-namestniški strežnik (DHCP proxy server). Ta lahko poskrbi za serviranje informacij, ki jih glavni DHCP-strežnik ne servira. Najlepše pri tem je, da ni potrebno ničesar nastavljati,

temveč le pripravimo in aktiviramo namestniški strežnik. Zanj potrebujemo program dnsmasq, ki ga instaliramo z ukazom:

```
sudo apt-get install
dnsmasq
```

Gre za preprost DHCP-strežnik, od katerega bomo uporabljali samo namestniški (proxy) del, ki obenem deluje tudi kot strežnik TFTP.

Predpostavimo, da ima naš strežnik IP-naslov 192.168.2.20, RPi pa bo od DHCP-strežnika dobil naslov 192.168.2.210.

Najprej naredimo rezervno kopijo stare datoteke /etc/dnsmasq.conf, nato pa vanjo vnesemo (komentar # lahko izpustite):

```
# potrebujemo samo proxy strežnik:
port=0
dhcp-no-override

# MAC naslovi RPi se začenjajo z B8:27:EB
dhcp-mac=set:rpi,b8:27:eb:*. *.* *
```



```
# ostali naslovi niso RPi
tag-if=set:notrpi, tag:!rpi

# potrebujemo še TFTP strežnik in povedati njegov korenski direktorij
enable-tftp
tftp-root=/var/lib/tftpboot

# na katerem omrežju operira dnsmasq, naj bo kar IP dnsmasq strežnika
dhcp-range=192.168.2.20, proxy

# za rpi imamo datoteko bootcode.bin na TFTP strežniku 192.168.2.20
pxeservice=tag:rpi, bootcode.bin, 192.168.2.20
# če hočemo imeti datoteke za vsak rpi posebej,
# odkomentiramo spodnjo vrstico:
# tftp-unique-root
# datoteke za vsak RPi bodo tedaj v podmapi z IP naslovom
# npr: /var/lib/tftpboot/192.168.2.210
```

V mapo /var/lib/tftpboot skopiramo datoteke iz mape /boot na SD-kartici.

Popravimo prekopirani /cmdline.txt, da se bo namesto medija priklopil omrežni datotečni sistem. Po novem naj cmdline.txt izgleda takole (vse mora biti v eni sami vrstici):

```
dwc_otg.lpm_enabled=0 console=serial0,115200 console=tty1 root=/dev/
nfs nfsroot=192.168.2.20:/var/lib/nfs/rpi210,udp,vers=3 elevator=deadline
rootwait ip=dhcp smsc95xx.turbo_mode=N
```

NFS-strežnik

Najprej namestimo strežniški paket in naredimo mapo /nfs/rpi210, v kateri bodo datoteke za RPi na naslovu 192.168.2.210:

```
sudo apt-get install
nfs-kernel-server

sudo mkdir -p /nfs/
rpi210
```

V to mapo prekopiramo vsebino korenskega imenika SD-kartice (druge particije, mmcblk0p2):

```
cd /var/lib/nfs/rpi210

sudo cp -rav /media/
admin/xxxxxxxx-xxxx-
xxxx-xxx/* ./
```

Mapa /media/admin/xxxxxxxx-xxxx-xxxx-xxx je tu mapa, v katero je priklopljena druga particija SD-kartice.

Pripravimo mapo za uporabo (ena vrstica):

```
echo "/var/lib/nfs/rpi210 *(rw, sync, no_subtree_
check, no_root_squash, insecure)" | sudo tee -a /
etc/exports
```

Restartamo storitev nfs:

```
systemctl restart nfs
```

Naredimo še prehode v požarnem zidu za potrebne storitve:

```
firewall-cmd --permanent --add-service=nfs
firewall-cmd --permanent --add-service=mountd
firewall-cmd --permanent --add-service=rpc-bind
firewall-cmd --reload
```

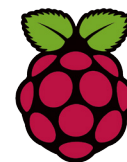
Ob težavah

Najbrž jih ne bo malo. Preden sem spravil skupaj tale članek in da je vse zares delalo, sem potreboval skoraj cel teden. Zato bom dal še nekaj nasvetov:

- Priporočam, da si distribucijo OS za RPi pripravite na SD-kartici in na USB-ključku, da ju boste lahko izmenjevali, ko bo šlo kaj narobe. Postopek je natanko isti. Poskuse sem delal na distribuciji Raspian Jessie (2017-02-16-raspbian-jessie).
- Datoteke lahko namesto s SD-kartice kopirate s ključka, kar je še posebno v redu, če nimate pri roki čitalca kartic.
- Pri eksperimentiranju nalaganja OS z omrežja lahko sistem prisilite k nalaganju z omrežja tako, da vanj vstavite FAT32-formatirano kartico, na kateri je ista datoteka bootcode.bin kot na strežniku. Pomagalo naj bi tudi pri zunanjih diskih, vendar tega ne morem potrditi.
- Ne delujejo vsi usmerniki in preklopniki enako; pri nekaterih ne bo problemov, drugod bo brez zgoraj omenjene kartice uspešen le vsak deseti poskus.
- Imejte dobro mero potrpežljivosti. Dokumentacija na spletu je eksperimentalna, nepopolna in ponekod tudi zastarela.
- Ukaz sudo reboot bo včasih namesto ponovnega zagona onesposobil sistem in bo treba RPI resetirati ali ga izklopiti. Prazna pina z oznako RUN sta namenjena reset tipki ...

Za konec

Vse opisano v tem članku je potrebno vzeti z določeno mero rezerve. Stvari programske še niso povsem dodelane in se lahko sčasoma pošteno spremenijo. Lahko tudi, da nikoli ne bodo popolne. Na internetu boste našli tudi druge metode nalaganja operacijskega sistema, ki so jih razvili v skupnosti in so trenutno bolj zanesljive od opisanih.



Viri:

<http://www.makeuseof.com/tag/make-raspberry-pi-3-boot-usb/>
<https://forums.plex.tv/discussion/254685/raspberry-pi-3-raspbian-nfs-plex>
<https://www.raspberrypi.org/forums/viewtopic.php?f=28&t=38325>

PIFcamp, mednarodni hekerski tabor v Trenti

Na 2. PIFcampu se je julija in avgusta 2016 zbralo okoli petdeset piflark in piflarjev z vseh koncev sveta, ki imajo radi naravo in hekanje programske in strojne opreme. Pomešali so umetnosti, znanosti in tehnologije, hkrati pa so se tudi zabavali.

Organizatorja tabora sta bila Ljubljanski laboratorij za znanost in umetnost – Ljudmila in Zavod Projekt Atol.

Svetlobni umetnik Tilen Sepič si je na lanskem taboru zamislil svetlobno kroglo Eclipse, ki zdaj

hiti po vseh svetlobnih festivalih. Intermedijska umetnica Robertina Šebjanič je na istem dogodku razvijala projekt Oscillatorium, s katerim se je potem predstavila na festivalu Kiblix.

Na letošnjem taboru so najbolj spodbujali projekte s področja e-tekstilij in nosljive elektronike, bioumetnosti, »naredi-sam« elektronike, računalniške umetnosti. V goste je prišla pionirka na področju nosljive elektronike in e-tekstilij Lynne Bruning. Ob obletnici potresa v Posočju so se nekateri lotili gradnje simulatorja potresa.

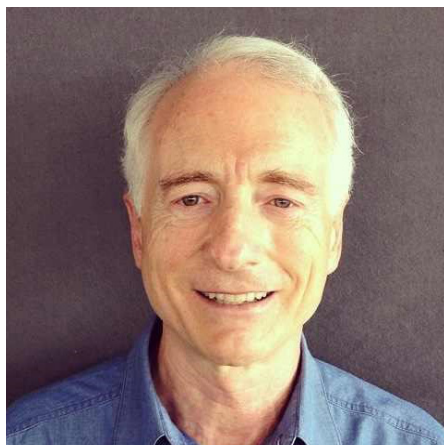
»Za začetnike je to prostor, kjer lahko dobijo zagon, v njem raziskujejo, dobijo ogromno znanja, kjer se lahko naučijo delati z rokami, pridobijo razne ročne spretnosti in obrtno znanje, tudi tako, ki izginja,« je razmišljala organizatorka Tina Dolinšek.

Udeleženci PIFcampa so zadnji dan tabora pripravili dan odprtih vrat in predstavili vse projekte, ki so jih razvili.

Vir: Mladina, 29. 7. 2016
(Petra Tihole, Piflarji v naravi).

Kopiraj in prilepi

Jernej Toth, G 2. B



Larry Tesler, pionir kopiranja in lepljenja

Vsakič, ko urejamo besedila, nam kopiranje lajša življenje in to vzamemo kot samoumevno. Velikokrat pa imajo ravno take stvari zanimivo zgodovino in ozadje.

Rezanje in lepljenje segata nazaj v čas rokopisov, ko so ljudje stavke

enostavno izrezali s škarijami in jih prilepili drugam. To se je ohranilo vse do danes, kar je bil navdih za naše najljubše ukaze.

Za vse to se lahko zahvalimo računalnikarju Larryju Teslerju, ki je bil zaposlen v podjetju Xerox PARC. Ukaze je dodal med letoma 1973 in 1976, ko je delal na grafičnem urejevalniku besedil Gypsy. Štiri leta kasneje pa je začel delati pri podjetju Apple in ukazi, kot jih poznamo danes, je Apple približal ljudem z računalnikoma Lisa in Macintosh.

Velika večina pozna kombinacijo tipke control in ostalih tipk. A operacijski sistemi Windows za kopiranje in rezanje uporabljajo manj znan IBM-standard. Za rezanje lahko uporabimo kombinacijo shift in delete, za kopiranje control

in insert in za lepljenje shift in insert. Izumitelj pa je postal, ker je postopek razdelil na dva dela. Prvi je rezanje ali kopiranje, ko naprava shrani podatke, ki smo jih izbrali v začasen spomin (clipboard), in v drugem delu podatke vstavi na željeno mesto.

Tesler je v kasnejših letih imel zanimivo življenje, saj je delal še v podjetjih Yahoo! in Amazon. V zahvalo Larryju pa bom kopiral zadnjo poved.

V zahvalo Larryju pa bom kopiral zadnjo poved.

-<https://goo.gl/y7b5db>, Tesler, Larry (2012). »A Personal History of Modeless Text Editing and Cut/Copy-Paste«

-<http://www.designinginteractions.com/interviews/LarryTesler>

VR ali virtualno okolje v igrah

Ramo Lulić, G 2. A

V zadnjih dvajsetih letih poskušajo vsa podjetja za razvoj računalniških in konzolnih iger razviti posebno vrsto doživetja za vse igralce različnih videoiger. To doživetje se imenuje VR (angl. Virtual Reality) ali virtualno okolje. To je okolje, v katerega igralec s svojim virtualnim likom vstopi in v katerem računalniško igo igra. Okolje ni resnično, temveč je imaginarno, računalniško programirano, grafično izpopolnjeno in praviloma obsežno, njegovi motivi pa so vzeti iz realnega okolja. V nekaterih primerih ti motivi obstoječi svet posnemajo

v tolikšnem obsegu, da postanejo njegova replika, medtem ko se v ostalih primerih lahko od njega bistveno razlikujejo. Virtualno okolje računalniške igre je torej lahko raznovrstno, to pa vpliva na stopnjo sprejemanja oglaševanja v njem.

V virtualnem okolju lahko igramo igre igranja vlog, kot so fantazijske igre, znanstvenofantastične igre ali večigralske spletne igre. Tako kot na računalnikih se je virtualno okolje razvilo tudi na mobilnih telefonih. Za igranje virtualnih iger je potrebno imeti

VR-očala ali slušalke, ki se prilagajajo igralčevemu gibanju med igro.

Evolucija VR-iger je dosegla najvišjo stopnjo leta 2016 in njihov razvoj se še ni ustavil. Zato so se pojavila vprašanja, kot sta: Ali bo to spremenilo prihodnost v smislu drugačnega doživljanja video iger? Ali bo to celo nadomestilo stvarnost z virtualno realnostjo? Na te odgovore bomo morali še malo počakati.

Vir: <http://www.cek.ef.uni-lj.si/UPES/mancini585.pdf> (17. 4. 2017)

Odslužene baterije vegovci uporabili v zmagovitem vozilu Foton

Dr. Uroš Breskvar



Ekipa Vegove z vozilom Foton. Voznika: Luka Hrastovec, Žiga Zajc

Mariborska fakulteta za elektrotehniko je 9. maja 2016 pripravila 5. tekmovanje z električnimi kolesi. Že lansko leto so se naši dijaki želeli udeležiti tekmovanja, pa kolo enostavno ni bilo dovolj dobro. Zato so za letošnje tekmovanje pripravili kar dve vozili. Obe sta tekmovali v odprti kategoriji, kar pomeni, da ni bilo omejitev glede moči motorja, vloženih sredstev ipd. Zato sta nastali zelo različni vozili. Prvo, poimenovano Foton, je videti kot avto v malem. Poganja ga 5-kW-asinhronski motor iz Letrike. Za karoserijo so dijaki uporabili avto Ci-

tyEl, ki je bil že prej električni avto. Od vozila je bila uporabna samo karoserija, pa še ta je bila v zelo slabem stanju. Dijaki so izdelali vso elektroniko, uporabili še krmilnik iz Letrike in vložili dobro leto dela.

Pri električnih avtomobilih je običajno največji strošek baterija. Prav pri tem so dijaki pokazali vso svojo iznajdljivost, saj so v podjetjih, kjer zbirajo odpadne baterije iz prenosnih računalnikov, zaprosili za takšne »izrabljene« baterije. Te so razstavili na celice, nato vsako izmerili in iz njih z enako karakte-

ristiko sestavili novo baterijo. Za to je bilo potrebno izdelati vezje, ki je ustrezno praznilo in polnilo celice in še beležilo rezultate za posamezno celico.

Rezultat? Prevoženih 82 krogov na hipodromu, kjer je krog dolg 800 m. Sledil je premik na lokacijo, kjer so vozila tekmovala v premagovanju klanca z 18 % naklonom. Vozilo je prišlo do vrha in štel je čas, potreben za vzpon. Po premiku v center Maribora so se tekmovalci pomerili še v spretnostni vožnji. In to vse z enim setom baterij – ostali

tekmovalci so baterije med etapami ali menjali ali polnili in v vozilu je ostala še vsaj polovica začetne energije. Uspeh je bil še toliko boljši, ker je imelo vozilo problem s pregrevanjem motorja, saj je vgrajen reduktor 1:4, idealen pa bi bil 1:8, za katerega pa žal ni bilo prostora za vgradnjo. Posledično je motor deloval z 2500 obrati na minuto, idealno pa bi jih bilo 5000. Ker motor ne deluje z optimalnimi obrati, ima slabši izkoristek in se segreva.

Dijaki so vsekakor dokazali, da je možno z ustreznim postopkom uporabiti že odpisane baterije.

Drugo vozilo je pa vsem zelo znani pony, ki so mu dali ime E-Pony. Vodila jih je misel, da predelajo slovensko kolo na elektriko. Enako kot pri fotonu so tudi zanj uporabili doma izdelane baterije. Prva je zadostovala za 60 krogov na hi-



E-Pony. Voznika: Kristjan Griljc, Klemen Kržič

podromu, pri čemer bi se lahko bolje odrezali, saj se v zadnji noči pred tekmovanjem baterije niso do konca napolnile zaradi napake avtomatike pri polnjenju. Motor moči 1000 W, ki je vgrajen v prvo kolo, je zadoščal, da je E-Pony brez ustavljanja prevozil klančino. Pri spretnostni vožnji je bil pa na mokrem cestišču kar preveč poskočen. E-Pony je zasedel 4. mesto in vsekakor pokazal, da bi bila lahko slovenska legenda ponovno oživljena v električni obliki in bi se odlično obnesel v mestnih središčih.

Elektrika v vozilu te vsekakor zasluzi, zato dijaki že snujejo nova vozila in se pripravljajo na nove izzive. Razmišljajo tudi, da bi svoje znanje uporabili za izdelovanje e-koles po naročilu, kar bi bilo zanimivo za ljudi, ki nimajo dovolj sredstev za nakup tovarniškega kolesa, sami ga pa ne znajo sestaviti.

Dijaki so s svojim znanjem in iznajdljivostjo vsekakor na pravi poti, da postanejo del uspešne električne mobilne prihodnosti.



Odslužene baterije iz prenosnih računalnikov



Odslužene baterije iz prenosnih računalnikov

Vašemu radijskemu sprejemniku so škodi dnevi

Nejc Bertonec, G 2. A

Digitalizacija medijev nedvomno prodira v naše življenje. V Sloveniji smo z njo začeli pri oddajanju televizijskega programa. Zdaj pa, po norveškem zgledu, sledi digitalizacija radijskih valov. Digitalni radio v tehnologiji DAB+ pri nas lahko že sprejemamo, a bo za množično popularizacijo potrebnih še nekaj let.

Na radijskem oddajniku Nanos so oktobra lansko leto slovesno zagnali digitalni radijski oddajnik. Predstavniki RTV Slovenija so na otvoritvi prehod opisali kot »pomemben razvojni korak« in Slovenijo umestili med »sodobno razvite države na področju radiodifuzije«, saj nova tehnologija DAB+ ponuja številne novosti in prednosti pred zdajšnjim, 84 let starim FM-sistemom (npr. boljša je kakovost zvoka, zanesljivejši je sprejem, konec je s prasketanjem in šumenjem, večja bo pestrost programov idr.). Sistem obljublja tudi optimizacijo na oddajniškem koncu, kot so manjša izhodna moč in posledično manjša poraba električne energije, učinkovitejša izraba frekvenčnega spektra in nižji stroški za izdajatelje vsebin. Sistem DAB+ prekaša internetni radio, saj je ločen od infrastrukture mobilnega omrežja, distribucija programa je veliko cenejša in naročnina za internet ni potrebna.

A sistem DAB+, čeprav se sliši kot popolna zamenjava zdajšnjega FM, le ne ponuja samo prednosti. Večina hišnih radijskih sprejemnikov ga še ne podpira, nakup novega, ki sistem podpira, pomeni dodaten strošek za gospodinjstva in posledično več elektronskih odpadkov,



saj tako FM- kot stari DAB-sprejemniki (predhodniki DAB+) ne bodo več koristni. Prav tako bodo zamenjavo potrebovali številni avtomobilski sprejemniki, saj je radio najbolj priljubljen ravno v avtomobilu. Na Norveškem, kjer trenutno pospešeno izvajajo »digitalni prehod«, se s problemom sprejema soočajo ribiči in motoristi, številni poslušalci, ki še nimajo novih sprejemnikov, pa ne morajo spremljati prometnih in vremenskih poročil; ta so za nepredvidljive norveške zime zelo pomembna. Oviro predstavlja tudi poraba energije, saj prenosni sprejemniki DAB+ izpraznijo baterijo občutno hitreje kot klasični FM-sprejemniki.

Zakaj pa smo se v Sloveniji šele zdaj odločili za prehod na digitalni radio, ko pa je bil v drugih državah (npr. Velika Britanija) v uporabi že veliko prej? Verjeli ali ne, začetki digitalnega radia v Sloveniji segajo že v leto 1997, ko so zagnali oddajnik na Krvavcu s takratno naj-

sodobnejšo tehnologijo DAB, a se sistem ni razvil po pričakovanjih in ni postal popularen. Danes, ko se za spremembo odločajo države, kot so Švica, Nemčija in Italija, pa smo se odločili za prehod tudi v Sloveniji. Za zdaj na digitalnih valovih zasledimo 13 radijskih programov, v nasprotju z Norveško pa lahko obstoječe radijske postaje še vedno poslušamo na FM-sprejemnikih.

Viri:

- Kričar (glasilo RTV Slovenija): DAB+, radio nove generacije (oktober 2016).
- <https://www.theguardian.com/world/2017/jan/11/norway-begins-switching-off-analogue-radio> (22. 1. 2017).
- <https://www.theguardian.com/media/shortcuts/2015/apr/20/oslo-calling-norwegians-worlds-first-digital-radio-fm> (+ fotografija, 22. 1. 2017).
- <http://www.wired.co.uk/article/norway-digital-radio-fm-switched-off> (22. 1. 2017).

Elektronska kocka

Vid Vrh, G 2. B

*V prvi letošnji številki revije **Elektror** sem pred nekaj časa zasledil zanimiv projekt, v katerem je eden od sodelujočih pri reviji naredil preprosto elektronsko kocko. Ker se mi je projekt zdel zanimiv, sem se odločil, da tudi sam poskusim izdelati podobno kocko.*

A še preden sem se lotil dela, sem se odločil, da načrt iz revije malo spremenim. Tako sem se odločil, da časovnik TLC555CP in desetiški števec 74HC4017 zamenjam z mikroprocesorjem ATTiny25V in tako prihranim 6 diod in nekaj prostora. Prav tako sem se odločil, da vsaka kombinacija ne bo imela svojega izhoda, kot je bilo predvideno v originalnem načrtu, ampak bom za vseh šest kombinacij uporabil le štiri.

Izdelavo kocke sem začel z načrtovanjem vezja. Za to sem uporabil program FreePCB, v katerem sem najprej moral narisati nekaj elementov, ki jih ni bilo mogoče najti v prednaloženih knjižnicah. A še preden sem se lotil dejanskega načrtovanja, sem moral določiti velikost vezja, in sicer zaradi omejenega prostora v ohišju, v katerega sem nameraval spraviti končni izdelek. Tako sem se znašel pred nalogo, da vse elemente spravim na vezje z maksimalnimi dimenzijami 52 x 53,5 mm. Glede na to, da sem želel, da končni izdelek izgleda kot kocka, sem moral upoštevati razporeditev LED-diod. Za te sem namreč želel, da so razporejene tako, da bodo, ko bodo prižgane, kombinacije izgledale tako kot na navadni kocki. Tu pa so se začele težave. Zaradi malo prostora

in velike množice povezav, nagnjenih okoli svetlečih diod v sredini, mi je vse elemente na ploščici uspelo povezati šele v četrtem ali petem poskusu.

Od tu naprej je bilo vse lažje. Ko je bil načrt končan, sem ga razmnožil in natisnil na poseben papir. Tega sem nato "nalikal" na bakreno ploščico, tako da se je črnilo prijeló ploščice. Papir sem nato previdno odstranil, tako da je ostalo samo črnilo. To je kasneje pri jedkanju služilo kot zaščita, ki je baker na mestih, kjer je bilo potrebno, zaščitila pred kislino, da ga ni odnesla s ploščice. Tako sem na koncu dobil vezje, ki sem ga kasneje še posrebril z odpadnim razvijalcem nekdanjih filmskih trakov. A še pred tem sem na mestih, predvidenih za tačke elementov,

izvrtal luknje. Za to sem uporabil sveder s premerom 0,8 mm.

Po izdelavi vezja sem se lotil ohišja kocke. Iz že obstoječega načrta vezja sem izbrisal žice, med oznakami za tačke LED-diod sem narisal križce in načrt znova natisnil. Nato sem ga nalepil na kovinsko ploščico, v katero sem na predvidenih mestih izvrtal luknje. Ploščico sem kasneje uporabil kot šablono, s katero sem si nato pomagal pri vrtanju lukenj v plastično škatlico. Za vrtanje sem uporabil sveder s premerom 3 mm, kakršna je tudi velikost LED-diod.

Ko je bilo ohišje narejeno, sem se lotil spajkanja. Najprej sem prispajkal upore, nato podnožje za mikroprocesor, sledil je gumb za vklop in nato še LED-diode in tipka. Pri spajkanju LED-diod sem moral upoštevati, da bi gledale skozi ohišje, in sem jih zato prispajkal na ustrezno dolžino.

Ko sem zaključil, sem se lotil še programiranja mikroprocesorja. Z osnovnim znanjem assemblerja sem napisal program, ki ob pritisku na gumb začne povečevati energo od registrov, in sicer do največje vrednosti 6. Ko je gumb ponovno pritisnjen, se štetje ustavi, prižgejo pa se tiste diode, ki skupaj tvorijo kombinacijo, ki bi na navadni kocki predstavljala vrednost registra.

S tako kocko bi bilo mogoče nadomestiti navadno kocko, kljub temu da je ni priporočljivo metati. Z njo se bi dalo igrati veliko večino (če ne celo vseh) iger, pri katerih je kocka potrebna.



Sistem za testiranje baterij

Luka Hrastovec, E 4. C

(Mentorica: mag. Maja Azarov Domajnko)

Vsako leto je veliko baterij za prenosne računalnike zavrženih. So res vse te zavržene baterije neuporabne in zanič [1]?

Sam sem se temeljito poglobil v testiranje baterij in v zadnjem letu intenzivno spoznaval razvoj postopka in sistema za testiranje celic. Moj testni postopek mi omogoča, da z zbranimi podatki s testiranja presodim, ali je celica še uporabna ali je dosegla konec svoje življenjske dobe in je čas, da se jo zavrže. Moja testiranja potrjujejo, da so celice še uporabne za baterijske pakete ultralahkih električnih vozil, kar lahko dramatično zmanjša strošek električnih koles za vse domače mojstre, ki se sami lotevajo izgradnje.

1. Postopek testiranja

Postopek testiranja je sestavljen iz petih različnih testov. Prvi korak je fizična ločitev vseh celic, ki jih dobim v baterijah za prenosne računalnike, kar je moj glavni vir za celice 18650.

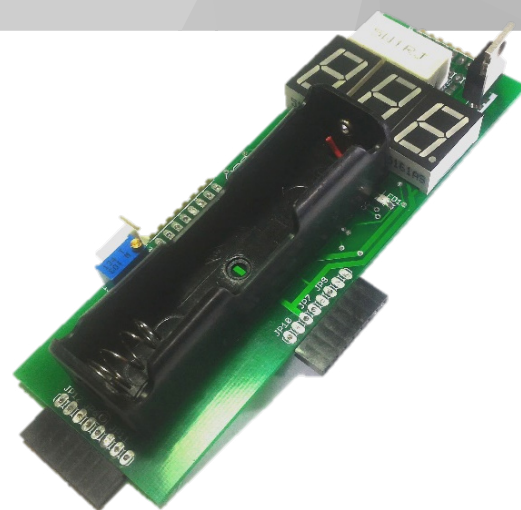
Baterije moramo odpreti s silo, saj nikoli niso bile namenjene razstavljanju in so običajno zlepljene skupaj. To je najtežji del v celotnem postopku, saj je delo fizično zelo naporno. Nekatere baterije imajo šibke točke in je odpiranje lažje. Sam sem odprl že preko 500 baterij, vendar je ta korak še vedno zelo zahteven in predvsem zahteva veliko časa.

Prvi test je vizualni pregled vsake celice. Ta test pokaže fizično stanje celice, saj je lahko poškodovana zaradi razstavljanja baterije ali pa iz nje izteka elektrolit. Vse fizično nespre-

jemljive celice zavržem in pošljem v reciklažo. Po pregledu vsaka celica dobi svojo številko. Po številčenju izmerim njeno prvotno napetost. Vse celice z neprimerno napetostjo zavržem in izbrišem s seznama. Takrat se začne pisanje podatkov v bazo. Zajeti želim čim večje število podatkov, kar mi kasneje omogoča, da lažje ocenim stanje posamezne celice. Vse celice s sprejemljivo napetostjo so pripravljene na polnjenje. Polnijo se s konstantnim tokom 500 mA, polnjenje ne sme trajati več kot 8 ur in celice se ne smejo preveč segreti. V podatkovno bazo zapišem čase uspešno napolnjenih celic. Po enem tednu jim ponovno izmerim napetost, ki jo tudi zabeležim v podatkovno bazo. Zavržem celice, katerih napetost je padla pod 3,7 V. S tem izločim tiste, ki imajo visoko stopnjo samopraznjenja. Od tega trenutka naprej je večina celic uporabnih. Tiste z majhno kapaciteto so sposobne samo manjših obremenitev. Ostale z večjo kapaciteto pa so primerne za pakete ultralahkih električnih vozil, kar je razlog, da celice razvrstim po njihovi kapaciteti, zato jih v zadnjem koraku testiranja ponovno izpraznim s konstantnim tokom 1 A. To nam pokaže njihovo pravo kapaciteto. Vse celice, ki so prestale opisane teste in imajo vsaj 80 % originalne kapacitete, so uporabne za baterijske pakete. Po praznjenju celice ponovno napolnim in jih hranim na ustreznem mestu, dokler niso na vrsti za uporabo v baterijskem sklopu.

2. Li4Li-sistem

Zaradi zapletenega postopka testiranja celic in težavnega sledenja sem



Li4Li master module

razvil Li4Li-sistem za avtomatsko testiranje [1]. Ime Li4Li pomeni "Lithium for Life". Ta proces ponovne uporabe celic je poznan tudi kot Lithium Second Life. Moj sistem opravi vse potrebne teste in še dodatnega, notranje upornosti celice. Sistem omogoča priklop 125 testnih modulov, kar omogoča istočasno testiranje 125 celic. Ta sistem predstavlja cenovno ugodno in celovito rešitev za vse domače mojstre, ki jih zanima izdelava električnega kolesa in baterije. Na trgu je nekaj rešitev za testiranje celic, vendar niso dovolj natančne ali pa so cenovno nedostopne domačim mojstrom. Povpraševanje po takem sistemu gotovo obstaja. Njegova glavna prednost je ta, da je natančnejši kot LiPo-polnilniki, ki se veliko uporabljajo za testiranja kapacitete celic.

Moj sistem je sestavljen iz strežnega modula in dveh različnih podrejenih modulov. Za strežni modul se trenutno uporablja Arduino Mega z Ethernet priklopom in čitalnikom za SD-kartice. Na strežnem modulu je dostopna spletna stran, ki služi kot uporabniški vmesnik. SD-kartico uporabljam za podatkovno bazo.

Prvi tip podrejenega modula je sposoben testirati le eno celico naenkrat, ki je uporabljen za testiranje kapacitete celice in je sestavljen iz pol-

nilnega vezja (TP4509), tokovnega bremena (LM358), mikrokrmilnika (ATiny85), zaslona in nosilca za celico. Mikrokrmilnik je namenjen komunikaciji in opravljanju različnih testov. Vsak strežni modul podpira do 125 modulov tega tipa.

Drug tip podrejenega modula je namenjen merjenju napetosti po enem tednu in merjenju notranje upornosti celice. Le en modul tega tipa je lahko priključen na strežni modul, saj test ni časovno potraten.

3. Osnovno delovanje sistema

Takoj, ko uporabnik vstavi celico v modul, ta prične s testiranjem. Najprej zabeleži originalno napetost celice in začne s postopkom polnjenja. Med polnjenjem se napetost celice beleži vsako sekundo. Zaključilo naj bi se v 8 urah. Če se celica v tem času ne napolni, se na modulu izpiše koda za napako. Po opravljenem polnjenju sistem čaka eno uro in med tem beleži napetost celice. Ta čas je potreben, da se napetost celice ustali in ohladi. Po ustalitvi sistem prične s postopkom praznjenja. Ta postopek traja od 30 minut do 3 ur. Čas je premo sorazmeren s kapaciteto celice. Celica se ponovno ohladi in po eni uri jo ponovno napolnim do 80 % glede na podatke iz prejšnjega testa. S tem je pripravljena na hrambo. Po tednu dni izmerim napetost in notranjo upornost celice.

4. Uporaba Li4Li-sistema

Moduli brez celic med delovanjem kažejo številko, ki je prosta. Vsakič, ko vstavimo celico v modul, nanjo

napišemo število, ki je izpisano na zaslonu modula. Ko je celica vstavljena, sistem takoj začne s testiranjem, kar traja približno 24 ur. Po končanem testiranju sistem pripravi celico na hrambo. Po enem tednu jo ponovno vstavimo, da se zabeleži napetost in izmeri notranja upornost. Po končanem postopku je v podatkovni bazi dovolj podatkov, da celice razvrstimo med dobre in slabe.

5. Moje ugotovitve in razvoj v prihodnosti

Na podlagi naših izkušenj delež uporabljenih celic niha med 20 in 80 %. Najboljše rezultate dobimo pri svežih baterijah. Ljudje jih puščajo na servisih računalnikov in te baterije so v relativno dobrem stanju, saj so sposobne napajati računalnik nekaj minut. Najslabše baterije so tiste, ki ležijo prazne na policah dlje časa. Danes lahko na trgu kupimo baterijo za električno kolo v razponu med 400 in 600 evri, odvisno od kapacitete in napetosti. Cena moje baterije je v razponu med 40 in 80 evri. To je približno desetina cene komercialno dostopne baterije.

V prihodnosti želim v svoj sistem implementirati nov strežni modul Raspberry PI. Njegova prednost je hitrost, boljša povezljivost z omrežjem, podpora za podatkovne baze in naprednejši uporabniški vmesnik.

Vira:

- [1] G. Pistoia and all, *Used Battery Collection and Recycling*, Elsevier Science, 2001.
- [2] A. Davide, *Battery Management Systems for Large Lithium Battery Packs*, Artech House Publishers, 2010.



Rabljene baterije prenosnih računalnikov



Razstavljena baterija



Skladiščenje celic

Razvoj gramofona in vinilnih plošč

Tin Pren, G 2. A

Začetki snemanja zvoka segajo v leto 1857, ko je francoski tiskar Édouard-Léon Scott de Martinville patentiral izum fonavtografa. To je najstarejša znana naprava, ki je bila sposobna snemati zvok, a posnetega ni mogla predvajati. Martinvillova naprava je služila kot navdih Thomasu Edisonu, ki je nato začel podrobneje raziskovati fenomen snemanja in predvajanja zvoka. Njegove raziskave so pripeljale do izuma fonografa leta 1877, ki je bil bolj napreden izum, saj je bil poleg snemanja zvoka sposoben tudi predvajati posnete vsebine. Naprava je delovala na principu zvočnih valovanj, ki so vplivali na nihanje membrane, na katero je pritrjena rezalna igla. Ta je na obod valja, prekrita s staniolom, zapisala vijačno sled v obliki vdolbin in izboklin. Pri predvajanju je druga igla povzročila nihanje membrane in pretvorila zareze v valju nazaj v zvok.

Prva oblika prodaje glasbe, primerljiva z današnjimi zgoščenkami, se je pojavila leta 1889, ko je Edisonovo podjetje izdalo posnete voščene valje za predvajanje, na katerih je bila posneta glasba.

Obdobje posnetkov na valjih in predvajanja s fonografi ni bilo dolgotrajno. Leta 1887 je Emil Berliner izdelal in patentiral prvi gra-

mofon. Menil je, da valj za zapis zvočnega posnetka nima prihodnosti, zato se je odločil za plošče kot nosilni medij. Prve so delovale pri sedemdesetih obratih na minuto in so imele premer okoli 18 centimetrov. Na njih je bila le na eni strani plošče zapisana spiralna sled – zapis zvoka. Postopek zapisovanja zvoka je bil zadovoljiv, vendar pa pri predvajanju ni dosegel kakovosti predvajanja s fonografi. V nadaljnjih raziskavah se je Berliner ukvarjal z izboljšavami postopka zapisovanja zvoka in leta 1892 patentiral izpopolnjeni gramofon. Ugotovil je, da je morala biti brazda v plošči dovolj globoka, s trdnimi robovi, da je sama vodila iglo po brazdi. Tako je dosegel, da je bil predvajani zvok s plošč podoben dejanskemu glasu. Berliner je hotel izboljšati tehnologijo množičnih kopij plošč. Leta 1893 je patentiral izdelavo cinkovih plošč. Iz njih je izdelal bakrene negative, s katerimi je tiskal plošče iz trdne gume in iz celuloida. Po letu 1897 je začel za odlitke plošč uporabljati posebne naravne smole živalskega izvora. Ta snov je omogočala glasnejše predvajanje posnetkov kot valji.

Vodilno vlogo na tržišču so prevzeli gramofoni. Bili so veliko preprostejši od fonografa in so jih zato lahko izdelali in prodajali za nižjo

ceno. Prve gramofone na ročni pogon so začeli prodajati okoli 1894. Leta 1896 pa se je na tržišču pojavil gramofon s pogonskim mehanizmom na vzmet s centrifugalnim regulatorjem hitrosti. Mehanizem tega gramofona je na Berlinerjevo pobudo razvil Eldridge Reeves Johnson. Zagotavljal je enakomerno in samodejno vrtenje gramofonske plošče. Johnson je razvil tudi postopek izdelave matric za tiskanje negativov iz posnetih izvirnih plošč, tako da zapiska na izvirniku ni bilo treba več fiksirati s kislino. Takó tiskane plošče so imele čistejši zvok in manj šuma.

Postopoma se je kvaliteta zvoka izboljšala in znižali so se stroški proizvodnje. Plošče, posnete pred letom 1902, niso imele glasbene vrednosti. V prvih letih 20. stoletja so že nastajali snemalni studiji. Začela so se snemati dela svetovnih zvezd glasbene oz. zabavne industrije. S posnetki teh nastopov je gramofon dobil svoj pomen in ga ohranil do danes.

Viri:

- D. Kunej in R. Kunej, *Glasba za obeh strani, Gramofonske plošče Mateja Arka in Hoyer tria. Ljubljana, Založba ZRC SAZU in Rokodelski center Ribnica, 2016.*
- M. Šetinc, *Viniliada.*
- Wikipedia.



Kondenzator za telebane

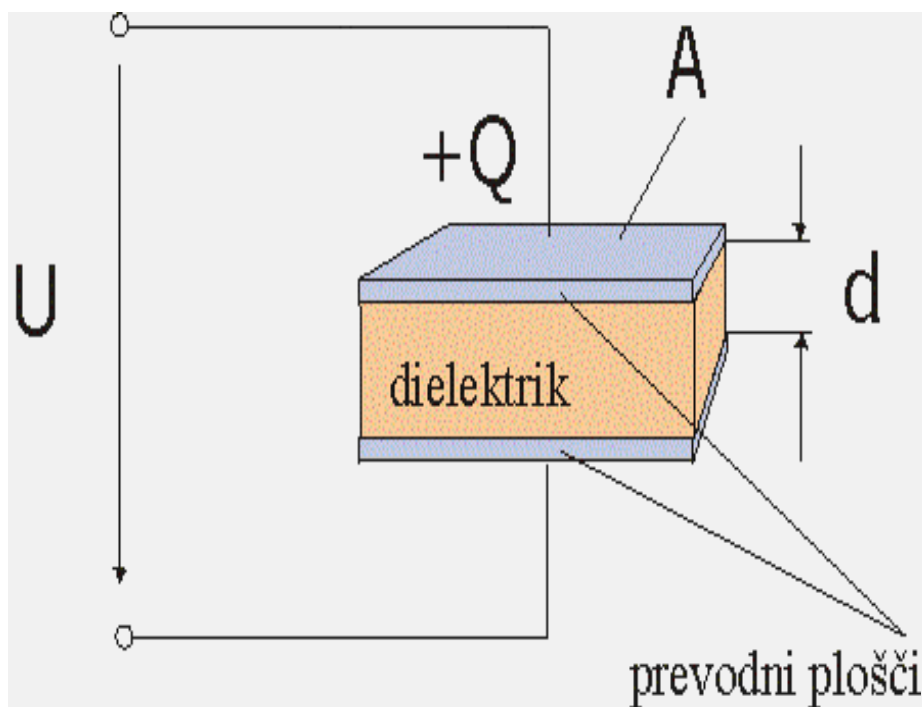
Jani Zrnc, G 2. A

(Mentor: Anton Orehek, univ. dipl. inž.)

Kondenzator je v elektrotehni-ki zelo uporaben elektrotehniški element, saj gladi nihanja v eno-smernem električnem vezju za proizvodjanje tokovnih ali nape-tostnih sunkov, shranjuje elek-trični naboj, vendar v nasprotju z ostalimi električnimi komponentami ne prevaja električnega toka.

Kondenzator torej shranjuje po-zitivni ali negativni električni na-boj (elektrino). Temu z eno bese-do pravimo kapacitivnost Q . Sam po sebi ne prevaja električnega toka, razen ko pride do električ-nega preboja dielektrika; v takem primeru pa je kondenzator neu-poraben. Prevaja ga le tedaj, ko se polni ali prazni (zelo kratek čas).

Kondenzator je v osnovi sestav-ljen iz dveh kovinskih plošč in neprevodne snovi med njima. Jakost električnega polja med ploščama, energija kondenzator-ja in na koncu tudi kapacitivnost so predvsem odvisne od razdalje med ploščama (d) in njune po-vršine (A) ter vrste dielektrika med njima. Včasih so bili ti le v trdnem agregatnem stanju, danes pa so tudi v tekočem, zato so tudi



Sestava kondenzatorja

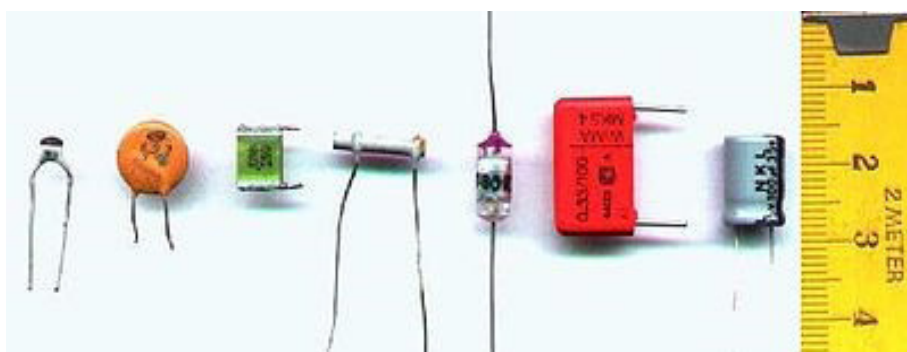
bolj zapleteni za izdelavo in imajo burnejše reakcije, ko pride do elek-tričnega preboja.

Poznamo različne vrste konden-zatorjev, kot so folijski, keramični, elektrolitski ... Danes največ uporabljamo elektrolitske kon-denzatorje, saj imajo najučinkovi-tejši izkoristek in temu primerno dolgo življenjsko dobo.

V električnem vezju je kondenzator skorajda nepogrešljiv element. Lahko ga vežemo zaporedno, vzpored-no ali pa kombiniramo ti dve vezavi in tako dobimo delilnike napetosti, različne kombinacije pa lahko tudi uporabimo, da dosežemo željene kapacitivnosti. Predvsem nepo-grešljivi so v televizijah oz. radijih, pa tudi v računalnikih jih ne manj-ka. V naših elektronskih napravah so v velikosti od nekaj milimetrov do nekaj centimetrov.

Vira:

- http://www.educa.fmf.unilj.si/izodel/sola/2002/di/Hodak/Elektricni_kondenzatorji/Index.html (22. 2. 2017).
- <https://upload.wikimedia.org/wikipedia/commons/thumb/d/de/Condensators.JPG/400px-Condensators.JPG> (22. 2. 2017).



Nekaj primerov kondenzatorjev

So osciloskopi res dragi? Izdelava poceni naprave

Luka Maček, G 2. A

Osciloskop je nepogrešljivo orodje v vsaki domači delavnici in laboratoriju. Z njim odkrivamo vse, od slabih kondenzatorjev do kratkih stikov in delovnega cikla PWM-naprav. A večina navadnih smrtnikov si ne more privoščiti teh profesionalnih naprav, ki stanejo tudi do več tisočakov. Prav zato se pojavlja veliko cenejših digitalnih alternativ, ki stanejo od dvajsetih evrov do nekaj stotakov. Tu se pojavi vprašanje o kakovosti, natančnosti in zanesljivosti teh naprav. Je vredno zapraviti tistih nekaj desetakov za orodje, ki svojo funkcijo opravlja slabo? Kaj pa, če ga lahko naredimo sami?

Najpomembnejše je seveda treba vedeti, kako osciloskop sploh deluje. To je naprava, ki v realnem času izrisuje graf odvisnosti napetosti od časa. Prvi osciloskopi so za to uporabljali analogne katodne cevi, danes pa prevladujejo digitalni. Ti so, če razlago močno poenostavimo, le analogno-digitalni pretvorniki (ADC), ki nam prebrano napetost izrisujejo v grafu na zaslonu.

Začnimo z ADC-pretvornikom. Eden najcenejših načinov je uporaba zvočne kartice. Te lahko berejo analogne signale z relativno visoko frekvenco. Signal lahko prikažemo s programom Oscilloscope Emulator avtorja Jerobea Fendersona, ki ga lahko brezplačno prenesemo s <https://asdfg.me/osci/>. V osnovi je sicer namenjen predvajanju audio datotek, a se da kot vir nastaviti tudi zvočni vhod. Naprednejša alternativa je program Soundcard Oscilloscope

Christiana Zeitnitza, ki je prav tako brezplačen na https://www.zeitnitz.eu/scope_en.

A programske rešitve morda niso naša stvar. Radi bi nekaj oprijemljivega in robustnega. Nihče ne mara priklapljanja čudnih kablov na svoj računalnik. Obstaja še ena rešitev – arduino. Najcenejše med njimi dobimo že za nekaj evrov. Imajo ADC in serijski vmesnik in so zelo enostavni za uporabo. Kako se torej lotimo izdelave? Ta je preprosta. Najprej moramo pripraviti programsko opremo. Na računalniku potrebujemo okolje Processing (<https://processing.org/>) in gonilnik za pretvornik z USB na UART-protokol, ki je vgrajen na plošči. Uporabljajo čip CH340G, za katerega gonilnik najdete na <http://www.arduino.cc/files/windows8/CH341SER.zip>. Za programiranje arduina seveda potrebujemo arduino IDE, ki ga dobimo z uradne spletne strani <https://www.arduino.cc/>. Iz <https://craft.vegova.si/images/vegovec/17/dds/osciloskop.zip> prenesemo datoteko osciloskop.zip in jo razpakiramo v poljubno mapo. Tam najdemo datoteko oscope_arduino.ino, ki vsebuje program za Arduino ploščo. Odpremo ga z arduino IDE in prilagodimo nekaj parametrov. Vrednost "bool SIG_OUT" nastavimo na "true", da nam bo program na pin 13, na katerem je običajno tudi LED-dioda, pošiljal testni signal za preizkus osciloskopa. Hitrost serijske povezave povečamo na 115200 baudov, tako da spremenimo "Serial.begin(9600)" v "Serial.begin(115200)", kar pripomore k

večji odzivnosti. Lahko uporabite tudi višjo (ali manjšo) vrednost, a točka, kjer se začnejo motnje, je odvisna od strojne opreme. Ko končamo, lahko kodo naložimo z gumbom "upload" v zgornjem levem kotu. Če smo uspešni, morata po končanem nalaganju svetiti POWER in TX-indikatorja (slednji v resnici le zelo hitro utripa, zato morda izgleda šibkejši), dioda L pa mora utripati dvakrat na sekundo, skladno s testnim signalom. Če se želimo prepričati, ali program deluje, lahko odpremo meni "Tools" in nato izberemo "Serial Monitor"; odpre se okno, ki naj bi prikazovalo niz na prvi pogled naključnih znakov. To so surovi podatki s plošče, ki pa jih je treba obdelati.

Za to poskrbi program "oscope_processing.pde", ki ga prav tako najdemo v zip-datoteki. Odpremo ga v okolju Processing. Tu moramo prilagoditi hitrost na enako, kot smo jo v programu za arduino, v našem primeru 115200 baudov. Nastavitev najdemo v ukazu "port=new Serial(this, Serial.list()[com_port], 9600);" v vrstici št. 77.

Zdaj lahko končno zaženemo program s trikotnim gumbom "run" v zgornjem levem kotu. Če imamo srečo, bi se nam moralo odpreti okno z grafom, ki prikazuje trenutno napetost na analognem pinu A0. V nasprotnem primeru se nam okno odpre, a obstane. Nad konzolo se nam v rdečem okvirčku izpiše sporočilo "Error opening serial port ... Incorrect serial port", ki nam poroča o težavah s serijsko pove-

EKG-ping pong

Janez Miklavc, univ. dipl. inž.



zavo. Težavo odpravimo tako, da dodamo ukaz "printArray(Serial.list());" nekam v funkcijo "setup", torej med vrstico 76 in 84. Ta nam bo ob zagonu programa v konzoli ob spodnjem robu izpisal oštevilčen seznam serijskih vrat, ki jih ima računalnik na voljo. Poskusimo večkrat, s priključenim in izključenim arduinom, in poskušamo ugotoviti, katera zaporedna številka (zapisana v oglatih oklepajih) mu pripada. To zapišemo namesto številke 1 v ukazu "int com_port =" v

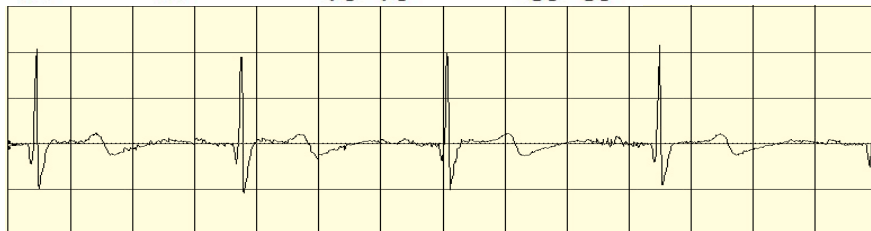
57. vrstici. Ko to popravimo, ponovno zaženemo program in, kot je v stroki navada, držimo pesti, da bo deloval. Ko vam ga uspe usposobiti, boste opazili, da se pojavi veliko motenj. Rešimo jih lahko z uporomo; sam sem uporabil 470 k Ω , ki ga povežemo med pina A0 in GND.

Čeprav naš izdelek deluje, nikakor ni nadomestek za profesionalni osciloskop. Pri nekaj deset kilohercih meritve niso več natančne. Poleg tega lahko meri napetost največ do 5 V, ki bo uporabna samo za kako analizo logike. Kljub temu je zabaven vikend projekt za začetnike ali pa malo bolj izkušene uporabnike, sploh za tako nizko ceno.

Viri (dostopni marca 2017):

- <https://asdfg.me/osci/>
- https://www.zeitnitz.eu/scope_en
- <https://processing.org/>
- <http://www.arduino.cc/en/files/other/ard-0008/CH341SER.zip>
- <https://www.arduino.cc/>
- <https://gist.github.com/chris-meyers-fsu/3270358>

Zanimiv izziv: izenačiti želimo srčni utrip para.



Če želimo, da imata osebi enak srčni utrip, kar pomeni 70 – 70 utripov/min ali 80 – 80 utripov/min, je najlažje uskladiti v prvem primeru z umirjanjem utripa druge osebe, npr. z vstopom v jacuzzi za 10 min. Lažji in ekonomični način pa je, da dvignemo utrip prve osebe s 70 na 80/min z aerobnim gibanjem, hojo ali tekom na mestu in ujemanjem skladnosti utripov. Kaj smo dosegli? Uskladitev srčnega utripa, vse ostalo je odvisno od nas. Če verjamemo, nam je pot odprta tudi v čustveno in mentalno uskladitev. Naši možgani in živčevje za prenos signalov plavajo za električne impulze v prevodni tekočini, srčna mišica pa se odziva nanje; izmerimo jih z EKG (elektro kardiograf). Za zajem EKG-signalov sem uporabil instrumentacijski ojačevalnik z zelo visoko vhodno impedanco (več kot 10 M Ω), za obdelavo in prikaz pa Sound Card Oscilloscope.

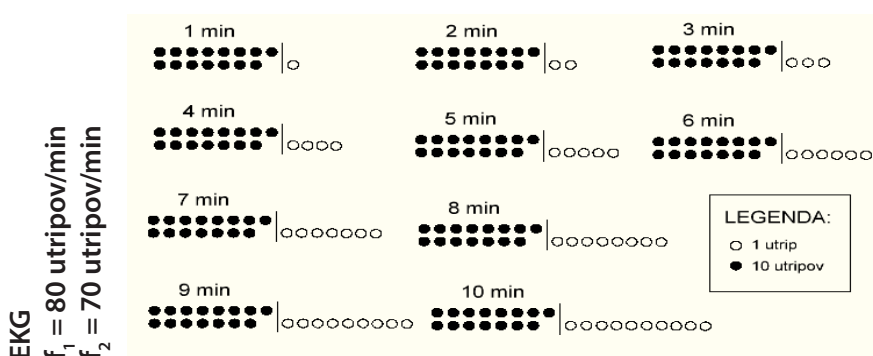
Hkratni utrip za telo pomeni:

- sočasen pritisk krvi v žilah in
- sočasno nasičenje kisika v možganih.

Tako nastopijo skladni pogoji sočasnega mišljenja, dela, čustvovanja, razumevanje drugega ...

Lahko se potrudimo ali pa se prepustimo spontani uskladitvi in bližini osebe vsakih 10 min, kar pa lahko tudi izračunamo z najmanjšim skupnim večkratnikom števil 70 in 80.

Ko vidimo mlade, polne idej, kako z iskro v očeh spremljajo z zanimanjem razne poskuse uporabnosti teoretičnega pouka v praksi, mi je to v največje zadovoljstvo in vesel sem za njih, da so zadovoljni, da želijo ustvarjati, da s svojimi idejami najdejo zanimivo, posodobljeno inovativno rešitev.



Vakuumske elektronke

Anton Orehek, univ. dipl. inž.

Uvod

Elektronke so predhodnice tranzistorjev. So (bile) osnovni gradnik elektronskih vezij (usmerniki, ojačevalniki, oscilatorji). Elektronska vezja so omogočila razvoj in napre-

dek radia, televizije in telekomunikacij. Prvi elektronski računalnik (ENIAC, 1946) je sestavljalo 17.468 elektronk. Prvi tranzistor je bil predstavljen 23. 12. 1947, prvi (delujoči) tranzistorski (radio) sprejemnik pa leta 1953. Dokončni zaton

elektronk se je zgodil v 70-ih letih prejšnjega stoletja, ko so v elektroniki že prevladovali tranzistorji, integrirana vezja in tudi mikroprocesor. V zadnjih dveh desetletjih so jih ponovno oživili v ojačevalnikih zvoka, ker imajo »boljše« karakteristike.



Vakuumske elektronke

Zgodovinski pregled

Prvo vakuumsko elektronsko cev (diodo) je leta 1904 izdelal John Fleming. Ta elektronka je imela le dve elektrodi: katodo (negativna) in anodo (pozitivna).

Leta 1907 je Lee DeForest vstavil »dodatno« elektrodo med katodo in anodo. Da je lahko prepuščala elektrone, je imela obliko mreže (angl. grid). Ta vakuumaska elektronka je znana kot trioda. Napetost med mrežico in katodo določa tok skozi elektronko in s tem omogoča krmiljenje toka.

Leta 1912 sta H. D. Arnold in Irving Langmuir ugotovila, da so električne lastnosti triod zelo po-

novljive, če je v elektronki »visoki« vakuum ($< 10^{-2}$ Pa). V istem letu je O. Wehnelt ugotovil, da je katoda bolj »učinkovita«, če je obdana z oksidno plastjo.

Leta 1915 je bila izdelana tetroda (elektronka z dvema mrežicama), ki je omogočala večje ojačenje, višje moči in tudi višje frekvence delovanja. Na karakteristiko tetrode je vplival efekt sekundarne emisije, kar je povzročalo nestabilnost in povečana popačenja pri visokih močeh.

Leta 1922 so odkrili postopek tesnjenja med steklom in vodniki (baker) in uporabili vodno hlajenje. Moči ojačevalnikov so narasle do 50 kW.

Leta 1919 so uporabili (iznašli) ločeno ogrevanje katode (uporabljeno šele 1927 z odcepom na transformatorju). Ogrevanje katode je v začetku potrošilo veliko (dodatne) moči, zato so precej napora vložili v učinkovitost ogrevanja in izbor oz. izdelavo oksidne prevleke katode (oddaja elektrone).

Leta 1926 je Bernard D. H. Tellen predstavil pentodo, ki je bila kasneje standardni ojačevalni element (do prihoda tranzistorja).

Kasneje so razvili še heksodo (1934) in heptodo, ki sta se uporabljali v radijskih sprejemnikih.

S tem je bilo konec elementarnega razvoja.

Kasneje so sledile izboljšave v zniževanju porabe pri ogrevanju katode, elektronke so postajale manjše, izboljšali so se proizvodni procesi, znižala se je cena. Spremenili in standardizirali so podnožja in napajanje za ogrevanje katode. Izboljšali so kakovost in učinkovitost delovanja katode, ki je bistvena za življenjsko dobo elektronke.

Na osnovi katodnih cevi so razvili visokofrekvenčne cevi velikih moči. Primerne so za oddajnike v območju UHF in mikrovalov.

Leta 1937 so razvili klistron, ki se še vedno uporablja za visokofrekvenčne oddajnike velikih moči.

Kasneje so izdelali tudi elektronko s »potujočim valom« (traveling wave tube – TWT), ki sicer omogoča nižje moči, a ima širše frekvenčno območje delovanja.

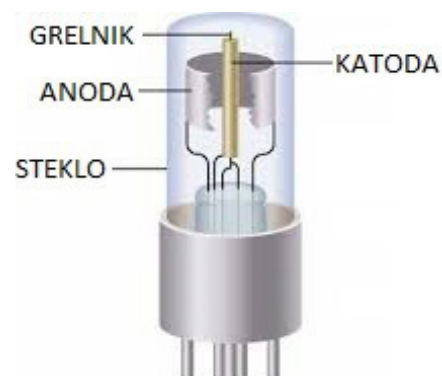
Delovanje vakuumskih cevi (elektronk)

Vse vakuumске cevi so zgrajene v stekleni posodi (kot žarnica), v kateri je »visoki« vakuum ($p < 10^{-2}$ Pa). Tak vakuum je potreben zato, ker med katodo in anodo letijo elektroni s hitrostjo, ki ionizira delce zraka na poti. Ionizacija spremeni električne lastnosti karakteristike, zato je nezaželena – škodljiva.

Dioda

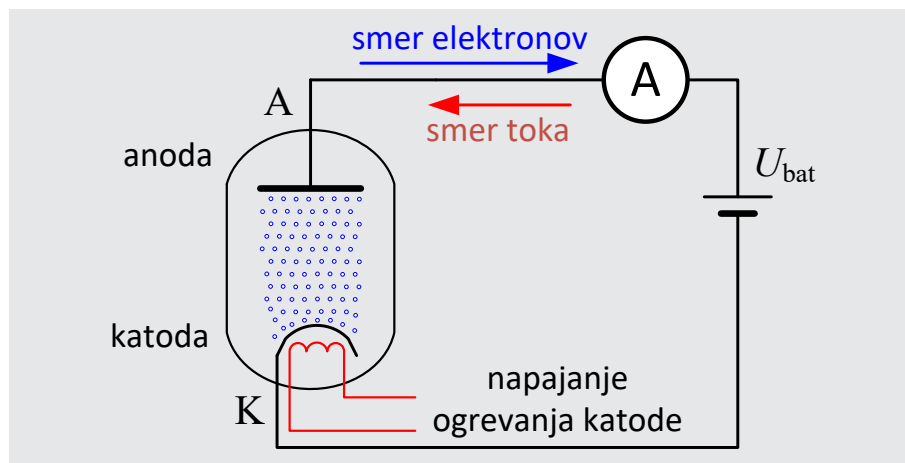
V bučki sta le dve elektrodi: katoda (K), ki jo segreva žarilna nitka (kot v žarnici), in anoda (A). Katoda ima visoko temperaturo ($>800\text{ }^{\circ}\text{C}$ za barij-aluminijev

oksid), zaradi katere iz nje »skačejo« elektroni (kot pokovka na vroči plošči), zato je obdana z oblakom elektronov. Če med katodo in anodo priključimo električno napetost (anoda je pozitivna), bo električno polje pospešilo elektrone v smeri proti anodi. V elektronki steče tok. Ker so elektroni negativni, je (pozitivna) smer toka od anode h katodi. Ker anoda ni segreta, tok v nasprotni smeri ne more teči (ni prostih elektronov). To pomeni, da dioda ne prevaja, če zamenjamo sponki napetostnega vira.

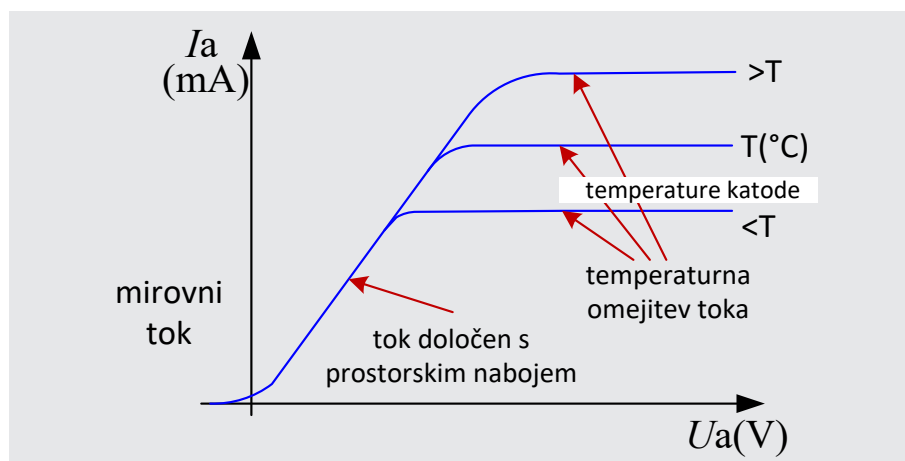


Vakuumska dioda

Pri vakuumski diodi tok narašča (dokaj) linearno z večanjem anodne napetosti. Tok je omejen s količino elektronov, ki jih določa temperatura katode (višja temperatura pomeni več prostih elektronov). Tokovi pri vakuumskih diodah so manjši od 1 A, so pa napetosti od nekaj 10 V do več 10 kV, zato so moči lahko visoke.



Shema električnega vezja vakuumске diode



Električna karakteristika vakuumске diode

Trioda

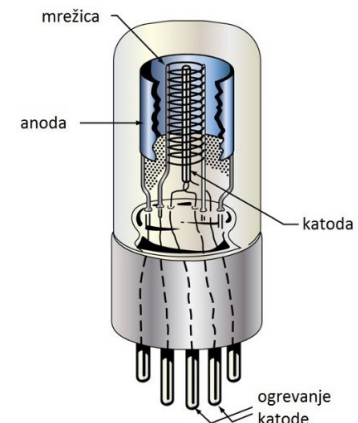
Trioda ima med katodo in anodo dodatno elektrodo, ki ima obliko mreže, saj mora skozi potovati snop elektronov (proti anodi). Ta elektroda se imenuje mrežica (angl. grid – g).

Mrežica se nahaja zelo blizu katode. Če med njo in katodo priključimo napetost, tako da ima mrežica nižji potencial, električno polje med mrežico in katodo ovira gibanje elektronov od katode proti anodi. Z napetostjo med mre-

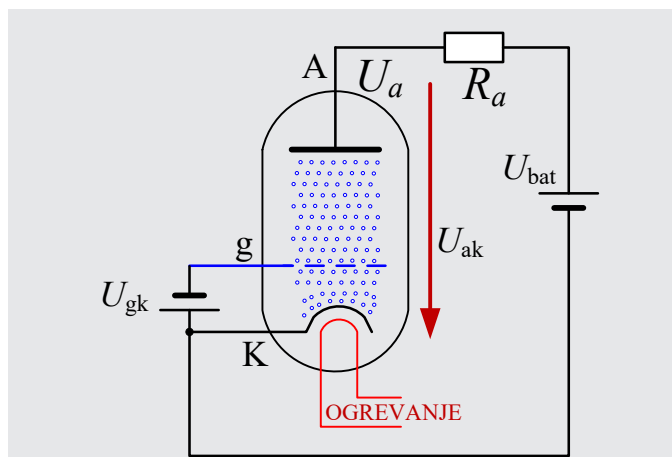
žico in katodo določamo pretok elektronov in s tem anodni tok. Mrežica deluje kot ventil, ki določa pretok vode. Ker je mrežica na negativnem potencialu, tok v mrežico ne teče.

Pri stalni napetosti med katodo in anodo je tok neposredno odvisen le od napetosti med mrežico in katodo. Tak element lahko deluje kot stikalo ali pa zvezno spreminja izhodni tok (napetost na bremenu). Ker za krmiljenje (napetost med mrežico in katodo) ne potrebuje (veliko) energije, trioda deluje kot

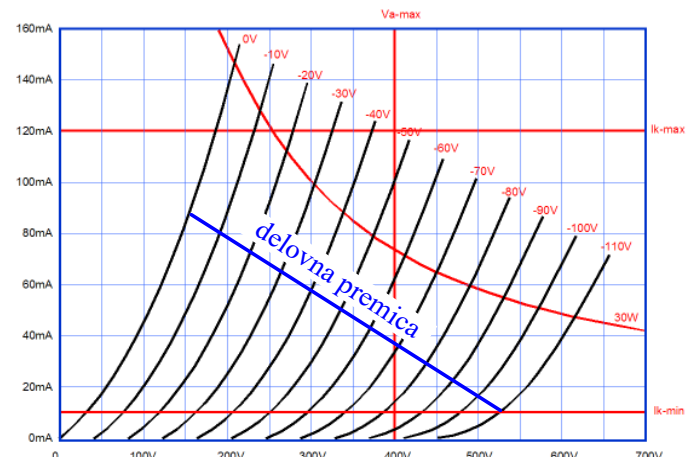
ojačevalnik električne moči (toka in napetosti).



Fizična zgradba triode



Električno vezje s triodo



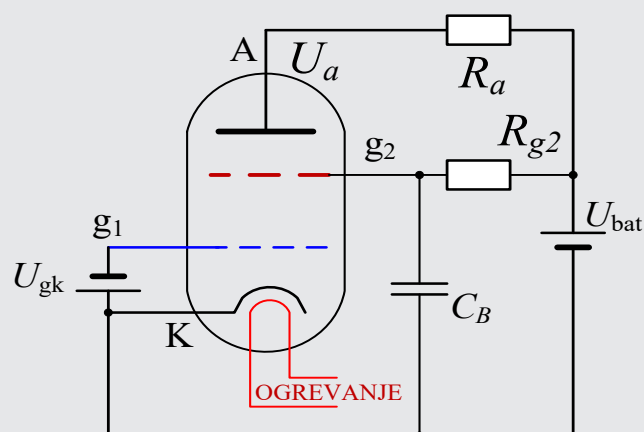
Izhodne karakteristike triode (2A3)

Tetroda

Tetroda je »trioda«, ki ima dodano (še) eno mrežico (g2) med prvo mrežico (g1) in anodo. Ta (dodatna) mrežica zniža vpliv spremembe napetosti med anodo in katodo na anodni tok. Ker je anodni tok skoraj neodvisen od napetosti anode, se ojačenje elektronke močno poveča. Mrežica (g2) je priključena na (stalno) pozitivno napetost in s tem pospešuje elektrone proti anodi. Ker je pozitivna, vanjo teče del katodnega toka. Velikost toka te mrežice je zelo odvisna od strukture (oblike) in

lege te mrežice. Ker se nahaja med prvo mrežico, na katero priključi-

mo krmilni signal, in med anodo (na njej je izhodni signal), se zniža



Električno vezje tetrode

kapacitivni sklop med izhodom in vhom, kar omogoča hitrejše delovanje (delovanje pri višjih frekvencah).

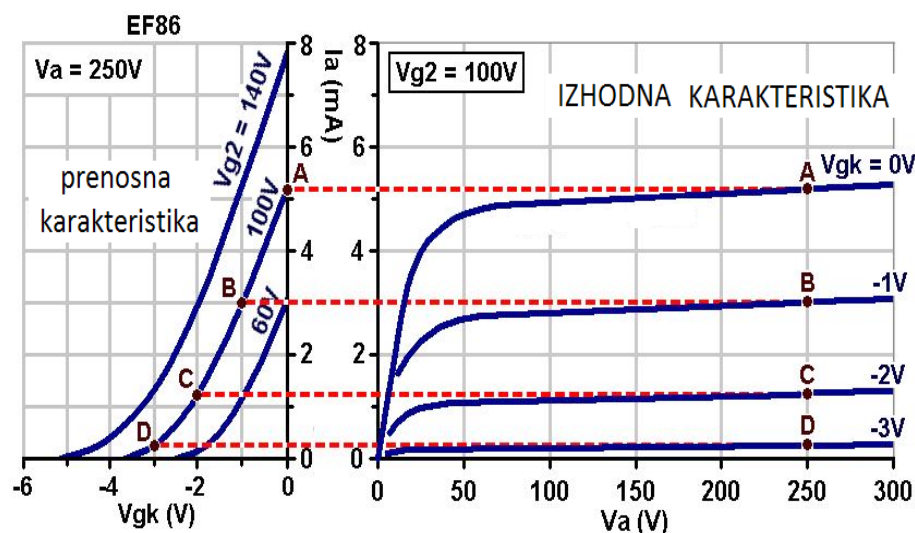
Tetroda je bila pomembna izboljšava, ki pa je pokazala slabost pri ojačevalnikih velikih signalov. Če je napetost na anodi nižja od napetosti na drugi mrežici, bodo sekundarni*1 elektroni stekli na mrežico g2. Zaradi tega se anodni tok zniža, to pa dodatno zniža izhodni signal, kar pokaže v popačenju signala.

*1 Sekundarni elektroni so tisti, ki jih primarni, ki priletijo iz katode (z veliko hitrostjo), izbijejo iz anode. Ti za hip izstopijo iz anode in nato padejo spet nazaj. Električno polje jih »vleče« v smeri elektrode z višjim potencialom.

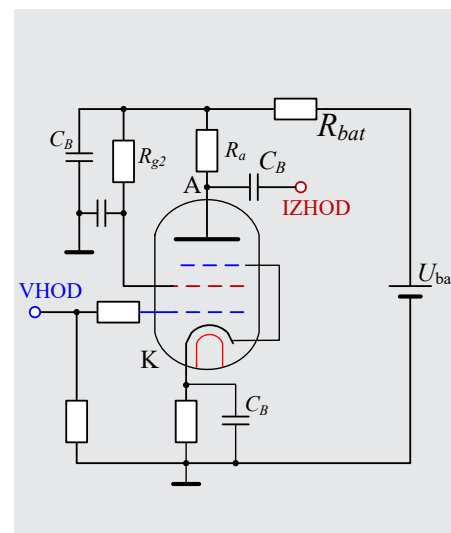
V določenem režimu delovanja je ob večanju anodne napetosti lahko anodni tok upadal. Tetroda ima v takih pogojih negativno upornost, kar je primerno za izdelavo oscilatorjev.

Pentoda

Pentoda ima tri mrežice oz. je to tetroda, ki ima dodano tretjo mrežico (g3), ki se nahaja tik pred anodo. Ta mrežica je električno priključena na katodo, zato je njen potencial vedno nižji od potenciala anode. Ta mrežica vse sekundarne elektrone usmeri (nazaj) na anodo in s tem odpravlja problem popačenj, ki jih ima tetroda. Pentoda ima (glede na triodo) višje ojačenje in boljše frekvenčne lastnosti. Pentoda je standardni ojačevalni element v močnostnih ojačevalnikih.



Vhodna in izhodne karakteristike pentode



Ojačevalnik s pentodo

Primerjava vakuumskih elektronk in tranzistorjev oz. polprevodniških elementov

Prednosti elektronk:

- linearno delovanje (brez povratne vezave), še zlasti pri majhnih signalih
- imajo »mehko« omejevanje amplitude (clipping), primernejše za avdio signale
- odporne so na motnje in napetostne impulze
- karakteristike so neodvisne od temperature okolice
- parazitne kapacitivnosti so majhne in stalne vrednosti (se ne spreminjajo s signalom)

Slabosti elektronk:

- so velike (veliko prostora), trošijo veliko moči (nekaj vatov samo za ogrevanje katode)
- potrebujejo visoke napajalne napetosti
- imajo slab (močnostni) izkoristek
- imajo majhen tok, zato tudi visoko izhodno upornost, potrebujejo prilagoditveni transformator
- so steklene in zato mehansko občutljive

- imajo življenjsko dobo do 5 (10) let oz. do takrat, ko se katoda iztroši (zmanjka oksidne plasti)

Prednosti tranzistorjev:

- imajo nižjo ceno kot elektronke, so precej manjši za enako moč
- lahko jih vezemo v vezja, kot so hibridna ali integrirana vezja
- imajo visok močnostni izkoristek
- delujejo pri nizkih napetostih (>1 V)
- odporni so na okolje in ravnanje

Slabosti tranzistorjev:

- imajo izrazito nelinearne karakteristike
- parazitne kapacitivnosti in upornosti se spreminjajo z velikostjo signala
- lastnosti se razlikujejo (celo) znotraj posameznega tipa in izvedbe tranzistorja
- lastnosti so odvisne od napajalnih pogojev (napetosti, tokovi) in temperature
- MOS FET ima visoko vhodno kapacitivnost
- ojačevalna vezja potrebujejo povratne vezave
- močnostni elementi potrebujejo (dodatna) hladilna telesa, kar vnaša nezanesljivost

Zaključek

Vakuumske elektronke so imele pomembno vlogo v razvoju elektronike, telekomunikacij in (celo) računalništva. V večini elektronskih naprav so jih nadomestili tranzistorji in integrirana vezja. V praktični rabi so le še močnostne katodne cevi, ki se uporabljajo za oddajnike v področju zelo visokih frekvenc (in moči).

Elektronke se (ponovno) uveljavljajo na področju avdio ojačevalnikov (še posebej so to kitarški ojačevalniki) in pa avdio ojačevalnikov visoke kakovosti (HiFi). Mnogi poznavalci trdijo, da je zvok, ki ga ojačuje ojačevalnik z elektronkami, »boljši« kot tisti, ki ga daje ojačevalnik s tranzistorji.

Sklepne ugotovitve

Električne sheme vezij z elektronkami so enostavnejša kot tiste s tranzistorji. To pa ne pomeni, da je izvedba določenih (namenskih) izvedb dejansko enostavnejša.

Pri polprevodniškem avdio ojačevalniku lahko uporabimo ojačevalni (hibridni) modul in kakovosten predojačevalnik in dvosistemski usmernik.

Pri ojačevalniku z elektronkami potrebujemo napajalnik in (dodatno) napajanje ogrevanja katod. Ojačevalnik z elektronkami potrebuje tudi (namenski prilagoditveni) izhodni transformator. Prav ta del (transformatorji) v veliki meri zmanjša »privlačnost« elektronk oz. zelo poveča problematiko (in ceno) izvedbe kakovostnega ojačevalnika (HiFi) z elektronkami.

Vsekakor je za »prave« avdiofile ojačevalnik z elektronkami primeren izziv. Več o izdelavi ojačevalnikov z elektronkami in o problematiki uporabe poiščite na spletnih povezavah, ki so navedene v zaključku zapisa.

Viri:

- <http://www.moah.org/education/transHistory1.html> (22. 12. 2016).
- https://en.wikipedia.org/wiki/Vacuum_tube (23. 12. 2016).
- http://www.vacuumtubes.net/How_Vacuum_Tubes_Work.htm (23. 12. 2016).
- <http://www.john-a-harper.com/tubes201/> (25. 1. 2017).
- <http://www.valvewizard.co.uk/index.html> (24. 12. 2016).
- <http://www.electrical4u.com/vacuum-diode-history-working-principle-and-types-of-vacuum-diode/> (24. 12. 2016).
- https://en.wikipedia.org/wiki/Tube_sound (25. 12. 2016).
- https://en.wikipedia.org/wiki/Valve_amplifier (25. 12. 2016).
- <http://www.philips-historische-producten.nl/tube-uk.html> (27. 12. 2016).
- http://www.mennovanderveen.nl/nl/download/download_3.pdf (28. 12. 2016).

- https://en.wikipedia.org/wiki/Traveling-wave_tube (28. 12. 2016).
- https://en.wikipedia.org/wiki/Hot_cathode (30. 1. 2016).
- <http://www.daenotes.com/electronics/devices-circuits/vacuum-tube-triode> (26. 12. 2016).
- http://www.valvewizard.co.uk/Triodes_at_low_voltages_Blencowe.pdf (26. 12. 2016).
- <http://www.valvewizard.co.uk/pentode.html> (26. 12. 2016).
- <https://www.ampbooks.com/mobile/tutorials/pentode-tutorial-2/> (26. 12. 2016).
- http://www.vias.org/crowhurstba/crowhurst_basic_audio_vol2_008.html (26. 12. 2016).
- <http://www.audioxpress.com/article/100w-triode-amplifier> (26. 12. 2016).
- <http://www.valvewizard.co.uk/pentode.html> (27. 12. 2016).
- <http://www.effectrode.com/signal-tubes/vacuum-tubes-and-transistors-compared/> (28. 12. 2016).

Delovanje triode:

- <http://www.r-type.org/static/grid01.htm> (27. 12. 2016).
- <http://www.lh-electric.net/tutorials/vt-prim.html> (27. 12. 2016).

Delovanje tetrode:

- <http://www.r-type.org/static/grid02.htm> (28. 12. 2016).

Delovanje pentode:

- <http://www.r-type.org/static/grid03.htm> (28. 12. 2016).

Analiza ojačevalnika:

- <https://www.ampbooks.com/mobile/tutorials/pentode-tutorial-1/> (30. 12. 2016).
- <https://www.ampbooks.com/mobile/tutorials/pentode-tutorial-2/> (30. 12. 2016).
- <https://www.ampbooks.com/mobile/tutorials/pentode-tutorial-3/> (30. 12. 2016).
- <http://www.john-a-harper.com/tubes201/> (25. 1. 2016).
- https://en.wikipedia.org/wiki/Hot_cathode (2. 1. 2017).

Robotika – Lego Mindstorms

Tomaž Platiše, G 2. B

Gotovo ste že slišali za robote. Ste se kdaj vprašali, kako robot sploh deluje oz. kako je sprogramiran?

To sem se tudi sam spraševal in zato sem se v osnovni šoli vpisal v krožek robotike, kjer smo delali z Lego Mindstorms, ki je poenostavljena različica za grajenje in programiranje robotov.

Najprej se je potrebno odločiti, kakšnega robota bi sploh radi naredili, saj vsak lahko počne drugo stvar in zato potrebuje različne sestavne dele. Ko se odločimo, ga lahko začnemo sestavljati po lastni ideji ali pa se držimo že narejene replike. Seveda ga je potrebno še velikokrat dodelati, saj vsakič manjka kakšen del ali senzor. Vedno začnemo z mini računalnikom, ki poganja celoten sistem, in

ne smemo pozabiti senzorjev, saj so robotova čutila in brez njih ne more delovati tako, kot bi moral. Ko ga dokončamo, se lahko lotimo programiranja. Preko računalnika ustvarimo program, sestavljen iz ukazov, ki jih robot nato poveže oz. ponavlja. Končni program pošljemo v robotov računalnik in ga zaženemo, da vidimo, kaj zna in kaj mu moramo še dodati. Tu pa se začne dolgotrajno delo z različnimi dopolnitvami v programu in na robotu, saj se ponavlja vsakič nekaj ne izide in je napako potrebno popraviti. Ko končno dela tako, kot bi moral, ga je prav lepo občudovati.

Dandanes se roboti uporabljajo že skoraj povsod. Vsak lahko naredi le nekaj, a so zelo uporabni predvsem

v tovarnah in tudi doma, saj jim nikoli ne zmanjka moči, razen če jih odklopimo od napajanja. Zelo dober primer je robotski sesalnik IRobot, ki čisti že veliko stanovanj. Njegov program pa je enostaven, saj gre samo za ukaz naprej in ko zazna oviro, se obrne za nekaj stopinj in program se nadaljuje. Seveda imajo novejšje verzije IRobota vgrajen že sistem za zaznavanje prahu, stopnic, ovir ...

Naredili so tudi že nekaj robotov, ki bi lahko delno nadomestili človeka, saj znajo kuhati, marsikaj prinesti, govoriti, peljati psa na sprehod, igrati nogomet in še marsikaj drugega, kar pomeni, da bo mogoče v prihodnosti lahko vsak imel doma robota, ki bo delal namesto njega.



Radijsko vodeni model oklepnega vozila

Luka Hrastovec, E 4. C

(Mentorica: mag. Maja Azarov Domajnko)

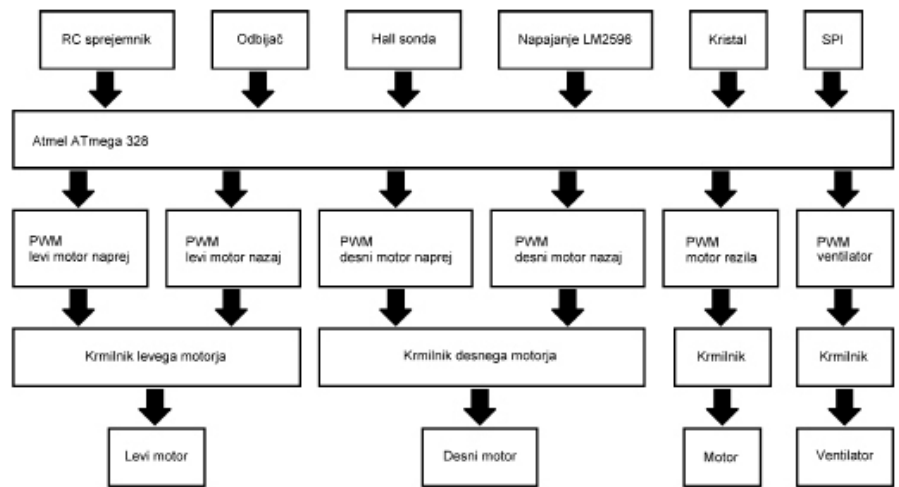
Internet je prostor, kjer veliko ljudi objavlja svoje projekte. Sam zelo rad raziskujem, kaj drugi izdelujejo v domačih laboratorijih ali garažah. Zelo rad tudi delam v svoji delavnici in potem objavim kakšen svoj projekt. Pred leti sem na spletu našel skupino ljudi, ki izdelujejo majhna oklepna vozila na daljinsko vodenje. Vozila imajo obliko tanka in so velika tudi do enega metra, večina je opremljena s kamero in brezžičnim prenosom slike. To omogoča vožnjo vozila iz naslonjača v dnevni sobi. Takoj sem vedel, da se moram takšnega projekta lotiti tudi sam, saj bi mi omogočal pluženje snega na dvorišču iz toplega naslonjača v dnevni sobi. Projektni teden v šoli sem zato izkoristil kot izgovor za izdelavo krmilne elektronike tega daljinsko vodenega modela.

Konfiguracija sistema

V prvi fazi sem izdelal blokovno shemo. V ospredju je AVR-mikrokrmilnik Atmel ATmega328. RC sprejemnik predstavlja vhodno enoto, dva krmilnika za motorje pa izhodne enote. Poleg tega sem dodal še nekaj senzorjev, ki pa v tem trenutku še niso implementirani.

Daljinski sistem

Za vožnjo tanka bom uporabil modelarski radio. To je oddajnik, namenjen vožnji radijsko vodenih modelov. Vsak modelarski radio vsebuje mikroprocesor, ki bere vrednosti štirih potenciometrov za pozicijo dveh krmilnih palic. Poleg dveh glavnih krmilnih pa-



Slika 1: Blok shema

lic ima še nekaj stikal in nekaj potenciometrov za poljuben namen. Mikroprocesor zajame vrednosti vseh vhodov in te vrednosti obdelava in jih pretvori v časovne intervale od 1 do 2 ms. Nato pa vse časovne intervale združi v en sam PPM-signal. Kratica PPM pomeni Pulse position modulation ali pulzno pozicijsko modulacijo. Vrednosti se pošiljajo s frekvenco skoraj 50 Hz, kar pomeni, da imamo čas ene periode malo večji od 20 ms. Čas ene periode se razlikuje od proizvajalca do proizvajalca. Največ jih uporablja za čas ene periode 22,5 ms, kar pomeni, da vrednosti osvežujemo s frekvenco 44 Hz. To nam omogoča pošiljanje 8 impulzov maksimalne dolžine 2 ms. Signal sprejememo s pomočjo sprejemnika, ki nam prej združene impulze razdeli in jih pošlje na izhode. Tako na sprejemniku dobimo PWM-sigale s frekvenco 44 Hz in obratovalni cikel (duty cycle) med 1 in 2 ms, odvisno od vrednosti.

Zajem podatkov

Te signale lahko z arduinom bremo na dva različna načina:

1. S pomočjo funkcije `pulseIn()`

Vsakič, ko kličemo funkcijo, ta počaka, da na vhodu dobi logično 1. Ko na vhod dobimo logično 1, funkcija začne meriti časovni interval do logične 0 na vhodu in to ponovi dvakrat. Nato funkcija vrne vrednost časovnega intervala v mikrosekundah. Slaba lastnost te funkcije je, da se ob vsakem klicu te funkcije ves program ustavi za vsaj 22,5 ms. Ker mora program brati dva signala, to pomeni 45 ms.

2. S pomočjo zunanje prekinitve

Lahko uporabimo funkcijo `attachInterrupt()` (glej vrstico programa 7 in 8). Ta funkcija nam nastavi vse registre za zunanje prekinitve. Ob vsaki spremembi se bo klicala funkcija, ki jo definiramo (glej vrstice programa od

Main.ino

```
7    attachInterrupt(digitalPinToInterrupt(2), interrupt1,
    CHANGE);
8    attachInterrupt(digitalPinToInterrupt(3), interrupt2,
    CHANGE);

23   void interrupt1(){
24       tank.values_update(0);
25   }
26   void interrupt2(){
27       tank.values_update(1);
28   }
```

Tank_drive.cpp

```
101  void Tank::values_update(boolean i){
102      if(time[i] > 0){
103          ch[i+1] = micros() - time[i];
104          time[i] = 0;
105      }else{
106          time[i] = micros();
107          overTime[i] = time[i];
108      }
109  }
```

23 do 28 in od 101 do 109). Tako lahko izmerimo obratovalni cikel signala.

S pomočjo obeh načinov lahko tako pridobimo dve vrednosti, ki sta v razponu med 1000 in 2000, in sta premo sorazmerni poziciji krmilne palice na oddajniku. Sam sem se odločil za drugo možnost,

saj sem napisal knjižnico, ki mi olajša pisanje programa za avtomatsko vožnjo v prihodnosti.

Obdelava podatkov

Ko dobimo podatke s sprejemnika, jih je potrebno obdelati. S sprejemnika dobimo 2 vrednosti za eno krmilno palico. Ena vrednost je za

```
35    PWMch[1] = constrain( map(ch[1], chPulse_min[1], chPulse_max[1], -255, 255), -255, 255);
36    PWMch[2] = constrain( map(ch[2], chPulse_min[2], chPulse_max[2], -255, 255), -255, 255);
37    PWMabs[1] = abs(PWMch[1]);
38    PWMabs[2] = abs(PWMch[2]);
39    backward_A = constrain(((PWMabs[1]-PWMabs[2]-133)*2),0, 255);
40    backward_B = constrain(((PWMabs[1]-PWMabs[2])*2), 0, 255);
```

odklon x in druga za odklon y. Obe vrednosti sta v razponu med 1000 in 2000. Ker bom na osnovi teh dveh vrednosti generiral PWM-signal, sem se odločil vrednosti pretvoriti v razpon med -255 in +255, ker ima mikroprocesor ATmega328P 8 bitni PWM. -255 predstavlja premikanje s polno hitrostjo nazaj ali desno in +255 premikanje s polno hitrostjo naprej ali levo. Za to pretvorbo sem uporabil kar funkcijo map() (glej vrstici programa 35 in 36). Prav tako v tem delu program izračuna še dve vrednosti, ki ju bom razložil malo kasneje (glej vrstici programa 39 in 40).

Vrednosti PWM-signalov glede na pozicijo krmilne palice

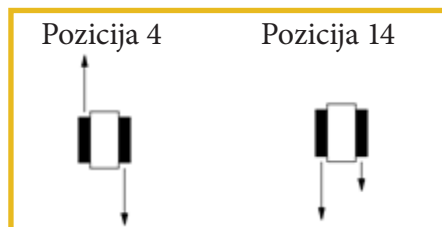
Področje krmilne palice sem razdelil na 19 različnih sektorjev. S stavkom if je določeno, v katerem sektorju se nahaja krmilna palica. Če pogoj drži, se v telesu stavka if kliče funkcija:

```
void Tank::Motors (byte
val_left_motor_forward, byte
val_left_motor_backward, byte
val_right_motor_forward, byte
val_right_motor_backward)
```

Ob vsakem klicu funkcije se vrednosti PWM-signalov na izhodu posodobijo glede na trenutni položaj krmilne palice.

Štirje posebni položaji se od ostalih razlikujejo in so bili narejeni za lepši prehod iz premikanja na mestu v vzvratno vožnjo. Na spodnjih grafikah vektorji prikazujejo hitrost motorjev. Prehod brez teh funkcij bi preveč obremenil tako

krmilnika motorjev kot mehanski prenos. V spodnjem primeru bi ob zelo majhni spremembi krmilne palice prišlo do velike spremembe obratov motorja, saj bi moral levi motor v trenutku spremeniti smer ob polnih obratih.



Generiranje PWM-signalov

Za generiranje PWM-signalov sem uporabil že kar vgrajeno funkcijo `analogWrite`. To funkcijo kličem iz svoje Funkcije `Motors`. Funkcija ima 4 parametre, ki so pomembni za krmiljenje motorjev. Odločil sem se za funkcijo, ker se jo lahko prilagodi za več vrst krmilnikov motorjev.

Zaključek

Moj projekt deluje, vendar kljub temu še vedno ni popoln. Program je napisan in je osnova, ki jo je potrebno razviti naprej. Med izdelavo projekta se je pokazala šibka točka mikrokrmilnika. Izbrani ni najbolj primeren za branje in generiranje PWM-signalov.

Veliko bolje bi bilo izbrati PIC-mikrokrmilnik, namenjen za pogonsko tehniko elektromotorjev. Na ta način bi bilo branje PWM-signala veliko lažje in natančnejše. S tem bi lahko dosegel večjo natančnost sistema, njegovo odzivnost in zanesljivost.

```

111 void Tank::Motors (byte val_left_motor_forward, byte val_
    left_motor_backward, byte val_right_motor_forward, byte
    val_right_motor_backward) {
112     analogWrite(left_motor_forward, val_left_motor_forward);
113     analogWrite(left_motor_backward, val_left_motor_
        backward);
114     analogWrite(right_motor_forward, val_right_motor_forward);
115     analogWrite(right_motor_backward, val_right_motor_bac
        kward);
116 }

```

Avtomobilček Arduino z bluetooth nadzorom

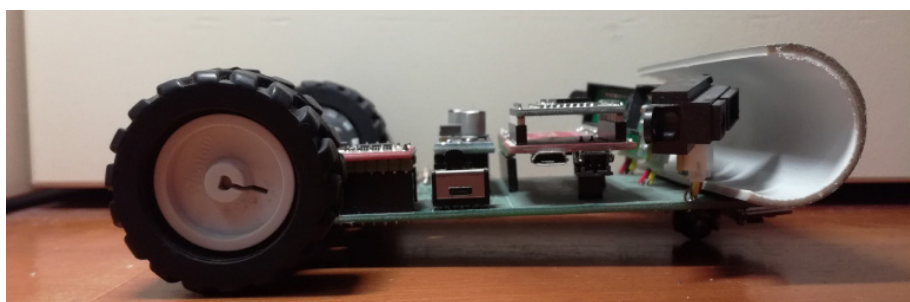
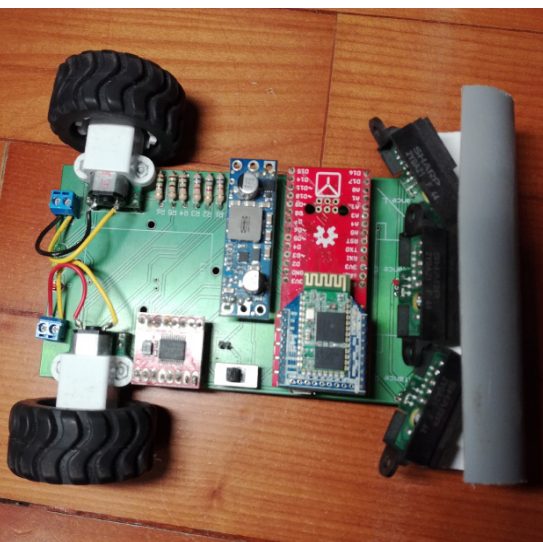
Matej Čampelj, G 2. A

To je avtomobilček, ki ga lahko sami sestavite in sprogramirate, tako da se bo povezal s vašim pametnim telefonom in ga boste lahko z njim usmerjali.

Premika ga mikrokontroler Arduino Micro, ki ga lahko programirate v okolju Arduino. Deluje s 4,5 V baterijo, ki se jo polni tako, da robota priklopimo na računalnik in ga prižgemo. Za premikanje uporablja dva servo motorja, ki določata smer robota glede na to, ali se vrtita naprej ali nazaj. Da lahko zadnji gumi usmerjata vožnjo, pa sprednjih koles ni, namesto njiju je krogla, ki leži na sredini sprednjega dela plošče.

Na sprednji spodnji in zgornji strani plošče pa ima senzorje za razdaljo in sledenje črte. Te omogočajo, da robot zazna ovire oz. se pelje po črti, ne da bi zavil z nje.

Za malo bolj nadarjene programerje pa je na robotu tudi čip BluetoothBee, s katerim se lahko povežete s svojim pametnim telefonom. Nato pa morate narediti le aplikacijo ali pa jo namestiti z interneta, da lahko kontrolirate premikanje svojega robota.



Robotika v zraku

Nejc Peciga, G 2. B

Leteči droni, ki jih dandanes uporabljamo za snemanje prireditvev, tekmovalij ali pa jih imamo le za zabavo, so bili uporabljeni že v drugi svetovni vojni, saj so nosili manjše pakete z eksplozivom in jih spuščali na sovražnike.

Od takrat pa je tehnologija že zelo napredovala in nanje so podjetja začela pritrjevati kamere ter raznorazne druge pripomočke, ki pa so se zelo dobro odrezali na raznoraznih področjih. Ker pa je v zraku veliko manj prometa (vsaj na taki višini, kjer potujejo droni), nekatera podjetja letalnike uporabljajo kar za dostavljanje naročil. Nekaj dražjih modelov pa ima v sebi vgrajeno tudi že pametno elektroniko, ki lahko letalnik samodejno pripelje nazaj do določene lokacije, ne da bi se pri tem zaletel v kakršnokoli

oviro, saj za to skrbijo vgrajeni senzorji. Kot pri vseh drugih pametnih napravah pa se tudi pri dronih pojavljajo hekrski vdori. Tako lahko nezaželjena oseba pridobi naše posnetke ali celo posnetke vojske, kot se je to zgodilo leta 2009 v Ameriki. Hekerji lahko prav tako prevzamejo kontrolo nad plovilom, če le to ni dovolj zaščiteno.

Ljudje pa se nismo zadovoljili le s tem, da bi iz zraka z droni le snemali, temveč, da bi z droni tudi potovali, vsaj na krajših razdaljah. Prav zaradi tega pa se je že razvil prvi električni dron, ki lahko nese enega človeka, težkega do okoli 110 kg. Uporabnik si sam nastavi pot in postanke, nato pa se usede vanj in se odpelje. Vožja ne sme biti predolga, saj lahko letalniku zmanjka baterije, uporabnik pa drona ne more upravljati, ko je v kabini. Takšna plo-

vila so zaenkrat še zelo draga, toda podjetja načrtujejo, da bodo ta vozila (plovila) že čez nekaj let služila tudi kot javni prevoz.

Tudi manjši kvadrokopterji spadajo v različne kategorije. Za njihovo upravljanje je potrebno v nekaterih državah imeti izpit določene stopnje, ki pa je odvisna od drona in dolžine poleta. Pri tem moramo biti pozorni, saj ob morebitni izgubi signala lahko koga poškodujemo, ali pa »le« izgubimo dron. Tudi zaradi te možnosti imajo nekateri že vgrajena padala, ki preprečijo hujše poškodbe osebe ali letalnika.

Dejstvo je, da se bo tehnika še razvijala, tudi v tej smeri in zanimivo bo videti, kako se bomo ljudje navadili na to.

Nova obzorja v svetu industrijske rabe dronov

Gal Štempihar, G 2. B

Drone ali brezpilotna daljinsko vodena zračna plovila že dolgo poznamo. Zadnjih nekaj let pa je veliko poučarka tudi na njihovi industrijski uporabi. Podjetja, kot so Amazon, Asda, BBC, že uporabljajo svoje drone na delovnih mestih. Dubaj se celo ukvarja s prevozom ljudi s pomočjo dronov, v Londonu pa se ukvarjajo z idejo prenašanja pošte s pomočjo dronov.

Podjetje DJI je razvilo nov dron, ki je namenjen industrijski uporabi na terenu. Podjetja bi s pomočjo drona lahko izvajala preglede različnih konstrukcij in druge terenske inšpekcije, s tem pa prihranila čas in denar.

Model M200 je izdelan za lete v težjih vremenskih pogojih, poleg tega pa uporablja ADS-B-sprejemnike, s katerimi zaznava druga zračna plovila, ki bi jim predstavljala potencialno nevarnost. Ti sprejemniki sprejemajo podatke o poziciji, višini in hitrosti drugih zračnih plovil v regiji, da lahko upravljalec drona primerno ukrepa. Domet drona je 7 kilometrov, v zraku pa lahko ostanejo do 38 minut. Droni M200 imajo tudi nosilnost do 2 kg in so opremljeni z najnovejšimi kamerami.

Čeprav so ADS-B-sprejemniki velik napredek glede varnosti, neka-

tere analitike skrbi, da bi bili zaradi takšnih varnostnih sistemov piloti preveč samozavestni in bi s tem ogrožali zračni promet, zato podjetje DJI namerava razviti tudi sistem proti trčenju dronov z zračnimi plovili s sodelovanjem podjetja Avionix, ki so specialisti na tem področju.

Vsi varnostni sistemi novih dronov še niso do potankosti razviti, vendar je to vseeno velik napredek v smeri industrijske uporabe dronov, ki bi lahko postali naš vsakdan že v bližnji prihodnosti.

Vir: BBC News.

Tehniško-naravoslovna pravljica

Rasto Snoj, prof.

Nekoč za devetimi gorami in vodami se je nahajala dežela Butale, a zastarele in že malce dolgočasne zgodbe o Butalcih so znane prav vsem, zato si oglejmo nekaj zanimivosti iz sodobne in neodvisne države Bananije, ki jo naseljujejo Bananci. Ti imajo v samem centru svojega najlepšega mesta, na katerega so sila ponosni, tudi prastaro šolo, ki pa je zamenjala že kar nekaj gospodarjev, sistemov in vizij. Nekateri so menili, da jo je treba zadnji hip dodobra prenoviti, vsaj tako se je odločil njen dolgotrajni voditelj, ki si je v skrbi za skupni dobrobit zadal še zadnjo odločilno nalogo ureditve zastarele razsvetljave. A kaj, ko je lahko tudi pot v pekel tlakovana z dobrimi nameni.

Ugotovljeno je bilo (to se v diplomatskem jeziku pač tako reče, da se elegantno izognemo neprijetnim podrobnostim), da je stara razsvetljava s po dvema 36 W General Elecrticsovima neonkama na sveto (fluorescenčnimi sijalkami, bolj učeno) potratna. Po domnevno bliskovitem premisleku (vse velike in pomembne ideje se rodijo v hipu, npr. jutri pišemo kontrolko, danes je še prezgodaj za učenje) je kocka padla, zamenjajmo torej stare potratneže z novimi LED-sijalkami z nominalno 63 W, to se pravi $2 \times 36 \text{ W} - 63 \text{ W} = 9 \text{ W}$ silnega prihranka na sveto, kar navsezadnje zna izračunati vsak vegovec z osnovno matematično kompetenco.

Seveda so LED v energetskem smislu vsaj še dvakrat učinkovitejše, tako jih reklamirajo, ampak



Živela prenova, stara krama stran!

šola jih je vseeno kupila natanko enako število, kot je že imela tistih 'zastarelih' neonskih svetilk. Hm, morda za rezervo - od viška glava ne boli. Monterji torej nemudoma prihrumijo in v pičlih 2 mesecih (izjemno hitro za Bananijo, če to primerjate z mega državnim projektom zmanjšanja čakalnih vrst v zdravstvu) postavijo enako število dvakrat učinkovitejših svetil. Že v času menjave in tudi kasneje sledi plaz jeremijad, glavobolov in neuslišanih pritožb prizadetih profesorjev pa tudi nekaterih dijakov, zato posledično celo odvisni 'strokovnjaki' ali po domače kar isti mojstri, ki so nam LED-sijalke prodali in zmontirali, priznajo, da smo zaradi tega (učeno rečeno) v fazi maksimalne prekomerne osvetljenosti. Kot pravi profesionalci nam seveda prijazno ponudijo 'sistemsko celovito rešitev = problem soluti-

on', ki se ji bolj po kmečko reče namestitvev dodatnih zatemnilnih (dimmer) stikal, da preveliko osvetljenost tudi 700 lx, čeprav za šole zadostuje 250 – 300 lx, vsaj malo zmanjšamo. Torej tako čez palec (odvisno od nastavitve zatemnilnega stikala) porabimo zdaj okroglih 72 W – 63 W/2 manj oz. okoli 40 W prihranka na sveto. Teh je v povprečnem razredu 9, takih razredov pa vsaj 15 (spet malce približno pa še zbornice in kofetarnice nisem štel, »prfoksi« naj kar trpijo). Torej gre za okoli $40 \text{ W} \times 9 \times 15$ ali okoli 5.5 kW prihranka pri električni moči.

Ne sliši se malo, ampak pogledajmo zadevo še malce drugače. Če si kupite povprečno toplotno izolirano okno s koeficientom toplotne prehodnosti $u = 1.0 \text{ W/m}^2\text{K}$ namesto takega, kot so tradicionalno, torej še iz časa Franca



Le umnim arhitektom cesarja Franca Jožefa se lahko zahvalimo, da imamo na Vegovi velika okna in dovolj zdrave dnevne svetlobe.

Jožefa na Vegovi (OK, nekatera so prenovljena, ampak le nekatera), boste dosegli večji energetski prihranek. Obstoječa okna namreč ob vsej preperelosti lesa, netesnjenju in ostalih lastnostih sodijo v kategorijo odpadne žx-kvalitete, koeficient u je vsaj $4 \text{ W/m}^2\text{K}$, kar zlahka preverimo tudi s termovizijsko kamero v IR-delu spektra. Tako pri npr. zunanji temperaturi -8°C in notranji 18°C (vegovci so kleni in že tradicionalno uživajo v mrzlih učilnicah) in pri povprečni površini enega okna 4 m^2 z zamenjavo prihranite $(4-1) \cdot (18 - 8) \cdot 4 \text{ W} = \text{okoli } 300 \text{ W}$. Razredi imajo ponavadi 3 ali več oken, se pravi 1000 W na razred, 15 razredov pa vam da 15000 W prihranka! Dobro, vendarle ne zaostajamo tako zelo, bodo za trjevali neugnani lučarski optimisti. No, pa to ni niti približno res, ker se šole ne da ogrevati le tedaj, ko so prižgane luči (npr. 2 uri na dan), ampak je treba to početi skoraj ves dan, kar prinese dodaten faktor razlike $24/2 = 12$. Z malo prizanesljivosti sklepamo,

da gre pri oknih **za vsaj 30-krat** tolikšen prihranek energije z zamenjavo.

Seveda pa se motite, če mislite, da je varčevanje poceni ne pri elektriki ne pri toplotni energiji. Varčevanje v javnem sektorju je v deželi Bananiji le drugo ime za metanje davkoplačevalskega denarja skozi okno, pač po načelu, kaj nas pa stane, če nismo tega mi sami prislužili s trdim delom. Po sicer nepreverjenih podatkih lokalne vohunske službe je bilo za našo svetlo prihodnost porabljenih kar 80 kEvrov . Pa se igravimo še z ekonomijo (kot veste, imamo pri nas tudi podjetništvo) in tole bo poskus male avtentične nalogice s sorodnega področja gospodinjstva.

Pustimo vendarle okna (mimo grede, obstajajo tudi subvencije EKO-sklada posebej zanje) in re-

cimo torej, da je pač 'prihranjenih' 5.5 kW ob porabi 2 šolskih ur. Seveda ne ves dan, ampak le tistih najbolj temačnih jutranjih, če bi seveda imeli prav vsi razredi pouk prve jutranje ure, kar zaradi dijakom prijaznega urnika ne drži, in lahko mi verjamete na besedo! To je torej energetsko isto kot $5.5 \text{ kW} \cdot 1.5 \text{ h}$. Šolska ura namreč traja 45 minut, če profesorji še malo zamudijo, s tem dodatno prispevajo k varčevanju z energijo, pa še toliko bolje. Zato to znese dobrih 8 kWh pri varčevane energije na dan. Če je cena 1 kWh energije 5.9 centov za male poslovne odjemalce (uradni podatki), to pomeni manj od pol evra na dan.

Da prav ste prebrali, gre za tisti rumeni kovanec, ki vam je včeraj padel skozi luknjo v žepu kavbojk! Če šola 'obratuje', npr. 10 mesecev po 5 dni na teden, to je



Strop temen, projekcijsko tablo osvetljujejo kar LED-sijalke, projektor je le še za okras ...

recimo dobrih 200 dni, je to 100 evrov letnega prihranka, seveda pa je (spet po zaslugi presvitlega Franca Jožefa) šola dovolj svetla z velikimi okni, in ko navsezgodaj posveti sonček, adijooo LED! Tako je prihranek zagotovo pod 100 ali celo 50 evri, res pa je, da cena elektrike v Bananiji ni cena elektrike, ampak so notri še omrežnine, davki, prispevki za energetska učinkovitost po 317. členu, prispevki za zagotavljanje podpor proizvodnji električne energije v soproizvodnji z visokim izkoristkom iz obnovljivih virov, trošarine in ostali ekonomski 'produkti', ki jih morate raznim prisklednikom plačati, da jo lahko sploh imate. Če torej zelo prizanesljivo ocenim vso zadevo, si upam trditi, da se bo ta investicija 'izplačala' čez morda nekaj desetletij ali celo kasneje.

Končno si oglejmo še drugo področje precej resnejših zadev, kot je denar. Nameščene LED-sijalke so v resnici model za skladišča (Philips Coreline Trunking LL121X LED75S/840), niso deklarirane kot priporočljive za šole (so torej neprimerne za šole), in to iz več razlogov. Definitivno so izjemno bleščeč, se pravi moteč vir svetlobe, kar je pogojeno z njihovo konstrukcijo. Tega dejstva niso zanikali niti vrli proizvajalci, ki seveda načeloma vedno branijo neoporečnost svoje robe, kakršna že je. V tem smislu LED ne pomenijo napredka, ampak nasprotno, nekaj korakov nazaj v primerjavi z 'zastarelimi', ki so bile vzorno nebleščeče. Drugačna postavitve (v obliki indirektna razsvetljave) bi to težavo odpravila, seveda bi bil potem svetlobni izkoristek

Toplotni tok $P[W]$ skozi npr. steno s površino $S[m^2]$ in debelino $d[m]$, s koeficientom toplotne prevodnosti $\lambda [W/mK]$ pri razliki temperatur med zunanostjo in notranostjo $\Delta T [K]$ dobimo kot

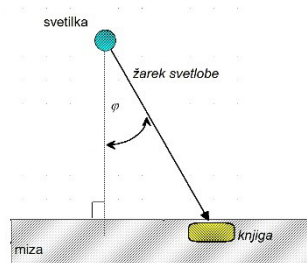
$P = \lambda \frac{\Delta T}{d} S$. Pri oknih tak zapis ni ravno praktičen, ker so uporabljeni različni materiali, več stekel, distančnikov, tesnil, lesen okvir zapletene oblike in zato pišemo prirejeno izvedbo zgornje enačbe kot $P = u \Delta T S$, kjer ima koeficient toplotne prehodnosti u enoto $[W/Km^2]$. Če je njegova vrednost recimo $0.9 W/Km^2$, to pomeni, da $1 m^2$ okna pri temperaturni razliki med notranostjo in zunanostjo $1K$ prepusti toplotni tok $0.9 W$.

Svetlobni tok P_s je energijski tok svetlobe v W , vendar človeško oko ne zazna vseh barv enako, zato je fiziološka enota zanj lumen (lm). Velja, da je $1W$ enobarvne rumenozelene svetlobe z valovno dolžino $560 nm$ ekvivalenten $680 lm$. Svetlobni izkoristek pa pove število lumnov oddanega svetlobnega toka na watt električne moči. Klasične žarnice imajo slab svetlobni izkoristek, ker oddajajo svetlobno energijo v širokem (predvsem v neškodljivem IR) spektru, je pa njihov izkoristek odvisen še od njihove električne moči, fluorescenčne svetilke imajo okoli $40 lm/W$, LED še več.

Osvetljenost površine E merimo v lux in je za točkast vir svetlobe (kar posamezna LED praktično je) določena z enačbo:

$$E = \frac{I}{r^2} \cos \varphi,$$

kjer I pomeni svetilnost, r je razdalja do npr. knjige na mizi, kot φ pa je kot med pravokotnico na ploskev in smerjo do svetila. Tako je lx isto kot lm/m^2 . Ploskev je najbolj osvetljena, če smo direktno pod svetilom, za vsako vrsto dela je vrednost predpisana oz. niso dopustne ne prenizke ne previsoke vrednosti.

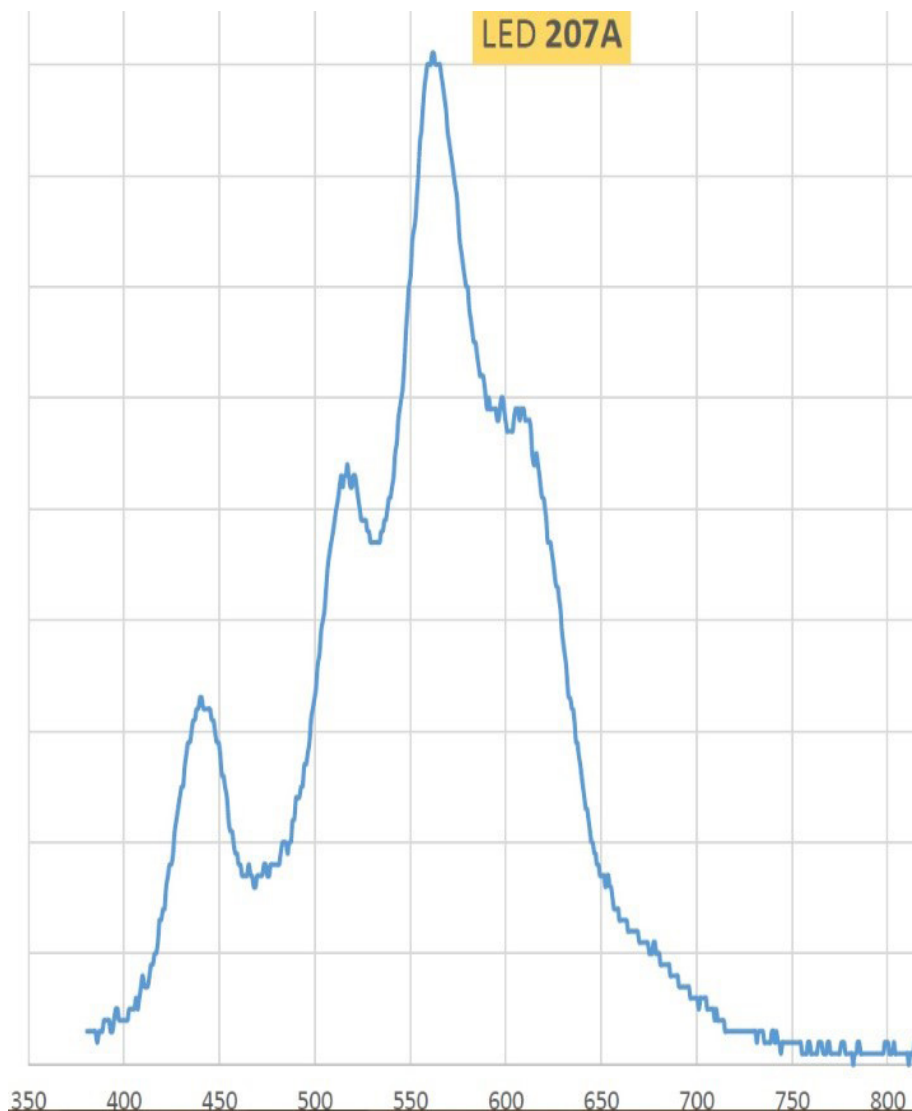


Problematično bleščavost B , bi nazorno lahko opredelili kot svetlobni tok P_s , ki ga v enoto prostorskega kota oddaja enota površine svetila. Motečo veliko bleščavost klasične žarnice z žarilno nitjo so že davno rešili s konstrukcijo večjega matiranega steklenega ohišja žarnice, tako ni bilo možno direktno videti tanke bleščeče žarilne nitke, pri fluorescenčnih ceveh pa je bila ta težava neznatna zaradi velike ploskve in tudi dodanih difuzorjev. Pri uporabljenih LED pa ves svetlobni tok direktno seva majhna ploskev LED, kar povzroči zelo močno bleščanje, k čemer doprinese tudi veliko število posameznih LED, ki jih je v nekaterih razredih tudi preko 1100!

Spekter svetlobe enostavno pove, koliko energije pripada svetlobi določene barve (valovne dolžine). Zaradi za določene barve različno občutljivih receptorjev-čepkov in palčic na mrežnici našega očesa je tudi odziv očesa na različne barve svetlobe drugačen. Pri neprimernem spektru veliki intenziteti so možne različne poškodbe vida, med drugim tudi degeneracija rumene pege, ki vodi v slepoto. Za posebej problematično velja (poleg UV svetlobe, ki jo običajne LED ne oddajajo) modra svetloba z valovno dolžino $440 nm$, ki je v spektru LED močno poudarjena.

precej slabši in upravičenost ekonomske investicije bi šla iz stotin v tisoče let. Pa še kakšnega arhitekta bi bilo treba povprašati (in plačati) za strokovno mnenje, saj elektromonterji le 'prešraufajo' in nič več. Močno bleščanje LED kot direktnega vira svetlobe povzroča tudi razne zdravstvene težave, je pa to tudi **eksplicitno prepovedano po veljavnih slovenskih pravilnikih**, ki urejajo razsvetljavo na delovnem mestu. Tudi dijaki, pa čeprav vegovci, delajo in šola je zanje delovno mesto, mar ne?

Neprimeren je tudi spekter s poudarjenim vrhom v področju intenzivne modre barve z valovno dolžino 440 nm (kar je sicer značilno za LED razsvetljavo) in je že dolgo časa znana kot zdravstveni hazard za vid. Slednje težave so delno odpravljive, če si nabavite očala s posebnim interferenčnim filtrom za natanko to valovno dolžino, kar vas bo stalo vsaj 200 evrov na lečo. Če bi vseh 900 dijakov to naredilo (pa še profesorji), bi bil strošek le zaradi tega 400000 evrov, kar je petkrat več, kot je bila vredna investicija v nove LED. Seveda ni rečeno, da boste te težave zares imeli, saj tudi kajenje ne pomeni, da boste dobili raka. Zadeva je še toliko bolj neprijetna, ker smo tudi sicer vedno bolj izpostavljeni taki svetlobi (pametni telefoni, LCD monitorji, televizorji, LED razsvetljava doma in v cestnih svetilkah, tudi v avtomobilskih žarometih ...). Nekateri tuje firme že uspešno tržijo ustrezno zaščito pod nazivom 'blue light protectors', tako za iPhone, LED-monitorje itd. Npr. Eyesafe (Health-E), Retina-



Spekter novih LED v eni izmed učilnic, zmerjeno s šolskim sistemom Vernier. Profesionalne meritve bi bile še bolj neprizanesljive. Na ordinati je valovna dolžina v nm, abscisa je v relativnih enotah. Količina energije med dvema valovnima dolžinama je sorazmerna s ploščino lika med tema in absciso. V rumeno-rdečem delu spektra (od 600 nm dalje) je te bolj malo, čeprav so LED nominalno bele.

Shield (Tech Armor), Frabicon, Cyxus, pri očalih pa Essilor Crizal Prevenzia, Carl Zeiss Dura Vision Blue Protect, itd. Ampak mi bomo o tem razpravljali kdaj drugič, morda tedaj, ko bo za široko rabo prodrla res kvalitetna OLED-razsvetljava, za katero pa je ta hip še prezgodaj, saj je treba najprej izprazniti skladišča, in to gre najenostavneje prav v Bananiji!

V okvirju na levi strani pa še malo fizike, sicer pa - veselo v svetlo prihodnost! Predvsem pa veliko sončnih dni. In ne pozabite na zaslužnega presvitlega Franca Jožefa.

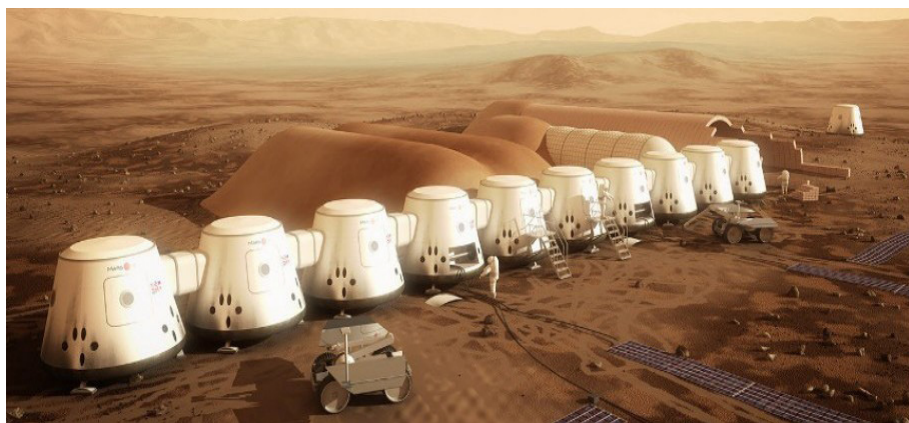
Ne pozabite, da je tole le pravljica. Morda je vse res, ali pa tudi ne. Bodite kritični, presodite sami.

Potovanje na rdeči planet

Ana Zobec, R 3. A

Elon Musk, ustanovitelj podjetja Tesla Motors in SpaceX in tehnološki vizionar, je leta 2011 na mednarodnem navtičnem kongresu (IAC) predstavil svoje načrte o kolonizaciji Marsa, do katerega naj bi z vesoljskim plovilom ITS (SpaceX) prispeli v zgolj 30 dneh. Prvi polet načrtuje do leta 2022. Za »letalsko vozovnico« na Mars bo treba odšteti 200.000 dolarjev. A Tusk svari. Pot bo nevarna in potniki morajo biti zanj pripravljeni umreti. Prvo plovilo, ki bo krenilo na Mars, bo nosilo ime Zlato srce (Heart of Gold) po izmišljenemu plovilu iz knjige Štoparski vodnik po galaksiji.

Na Marsu bo po Muskovi ideji v naslednjih 40 do 100 letih oz. 20 do 50 »medzvezdnih« voženj pozneje tako zaživela kolonija milijon ljudi. Sprva bo pot trajala 80 dni, pozneje pa nameravajo čas potovanja skrajšati na



Mars One: Prikaz prvega naselja na Marsu

samo 30 dni. SpaceX v ta namen izdeluje novo enostopenjsko raketo in vesoljsko plovilo ITS (Interplanetary Transport System), oboje narejeno v glavnem iz ogljikovih vlaken. Raketa bo imela premer 12 metrov, 49,5 metra dolga in 150 ton teže ITS pa 17 metrov, v višino pa bosta skupaj merila kar 122 metrov. Gnala ju bo jata že preizkušenih raketnih motorjev Raptor (delujejo na tekoči metan in tekoči kisik), in sicer raketo 42, plovilo pa 9. Ko se bosta raketa in plovilo ločila, se bo slednje začasno utirilo v Zemljino orbito, raketa pa se bo vrnila na izstrelitveno ploščad 39A v Cape Canaveralu na Floridi, na-

ložila tovarno plovilo z gorivom, ga utirila v orbito v bližini plovila ITS, s katerim se bo ta združil, »izpil« vse gorivo in »vesoljsko cisterno« poslal nazaj na Zemljo po nove avanturiste, ki že zdaj komaj čakajo na svoje potovanje na Mars. Za pridobivanje električne energije med potovanjem bo razprl veliki krili, ki bosta namesto v perje odeti v sončne celice z močjo 200 kW. Za pristanek na Marsu bo ITS spet aktiviral raketne motorje, svoje rezervoarje pa napolnil s tam pridobljenim metanom. Nato se bo vrnil nazaj na Zemljo po novih 200 potnikov. Plovilo naj bi pozneje krenilo tudi proti Jupitru in Saturnu.

»Quartz« kristali v Zemljinem jedru proizvajajo magnetno polje

Miha Faganel, G 2. B

Znanstveniki na Tehnološkem inštitutu v Tokiu (Tokyo Institute of Technology) so 23. 2. 2017 objavili raziskavo, v kateri poročajo o novem viru Zemljinega magnetnega polja, o vzrokih hlajenja jedra in njegovi kemijski sestavi ter o stanju Zemlje med njenim nastankom.

Jedro Zemlje je večinoma narejeno iz velike krogle tekoče kovine, ki se naha-

ja 3000 km pod površjem, in je obkroženo z vročim kamenjem. Ker je jedro tako globoko v Zemlji, je tam zelo velik pritisk in so visoke temperature. Vroče kamenje, ki obdaja jedro, se počasi premika, par centimetrov na leto, in s seboj prenaša toploto iz jedra na površje. V letu 2013 je Kei Hirose predpostavil, da se je jedro ohladilo za kar 1000 °C od nastanka pred 4.5 milijardami let.

Moralo bi se ohladiti za toliko, da bi vzdržalo Zemljino magnetno polje, razen če bi obstajal nov neodkrit vir energije. To ni bila edina težava, na katero so znanstveniki naleteli, saj tudi niso natančno vedeli, iz česa je narejeno Zemljino jedro. Znano je, da je jedro narejeno večinoma iz železa, nekaj niklja, da vsebuje približno 10 % lahkih zlitin, kot so silicij, kisik, žveplo, ogljik, vodik ...

Zemlji podobni planeti

Aleks Abdagić, G 2. B

22. februarja 2017 je NASA objavila raziskavo, v kateri so nas obvestili o odkritju 7 planetov v bližnjem sistemu TRAPPIST-1. Vseh 7 planetov ni bilo na novo odkritih, ker so za dva že vedeli maja 2016, a takrat niso dali nobenih novic, ker so sprva želeli preiskati celotni sončni sistem. Ta sistem leži v ozvezdju Vodnar in je od nas oddaljen približno 40 svetlobnih let, kar je v smislu vesoljskih razdalj še dokaj blizu.

Odkritje teh sedmih planetov je zgodovinska zadeva, saj se tako velika skupina Zemlji podobnih planetov do sedaj še ni pojavila v bivalnem delu sončnega sistema oz. v enem sončnem sistemu. Te planete je odkrila NASA in vesoljski teleskop z imenom Spitzer. Pri vesoljski agenciji že teoretizirajo, da ima vsak planet veliko možnosti za ključ do življenja in vode. Sistem TRAPPIST-1 ima »super hladno« pri- tlikavo zvezdo, ki oddaja le petstotino enega odstotka toliko energije kot naše Sonce, kar pomeni, da je bivalna

cona v tem osončju bistveno večja. Zaradi gostote planetov domnevajo, da so vsi kamniti in trdni kot Zemlja, kar pomeni, da si niso le podobni v velikosti, temveč tudi v zgradbi. Vsi planeti v tem sistemu so si zelo blizu, kar pomeni, da obstaja možnost, da se z enega planeta na nebu, kot je za nas Luna, vidi drugi planet. Iz dosedanjih opažanj so postavili hipotezo, da so planeti »plimno priklenjeni« na zvezdo TRAPPIST-1, kar pomeni, da planeti zvezdi kažejo isto stran in to ima lahko močan vpliv na vremenske razmere na planetih. Lahko so bolj ekstremne zaradi pojavov, kot so izredno hitri vetrovi in hitre spremembe temperatur.

Naslednji vesoljski teleskop, ki bo imenovan James Webb, bo lahko bolj natančno skeniral te planete v prihodnosti, saj bo imel sposobnost zaznave kemijskih odtisov v vodi, določanja sestave atmosfere, izmeritve gostote ozona in analize planetove temperature ter pritiska, kar so vse ključni

faktorji pri določanju ustreznosti za kolonizacijo v prihodnosti. Zaradi morebitnih ugodnih razmer za življenje na teh planetih znanstveniki menijo, da na enem od teh planetov obstaja neka oblika življenja. Astronomi vidijo odkritje teh planetov kot prehod v dobo razvite vesoljske tehnologije, s katero bodo dosegli več odkritij in izvedeli več o naši galaksiji ter mogoče še odkrili inteligentno življenje, s katerim bomo po možnosti lahko komunicirali. Sedaj se nam te stvari zdijo nemogoče in neuresničljive, a tehnološki napredek je že velikokrat presenetil javnost zaradi svoje spontanosti.

Odkritje Zemlji podobnih planetov je zelo pomembno za prihodnost človeške vrste, saj jih bomo mogoče nekega dne naselili. Čeprav planeti niso še bili v celoti analizirani, in še nekaj časa ne bodo, lahko predvidevamo, da bo vsaj eden ugoden glede na dosedanja opažanja, a se relevantnost teh eksoplanetov še dolgo časa ne bo zmanjšala.

Vodja raziskave Hirose je menil, da je od tistih 10 % prisotnih mnogo različnih zlitin. Nato so znanstveniki začeli eksperimentirati z možnimi zlitinami pri istih pogojih, kot so v jedru, da bi lahko ugotovili, katere kombinacije bi predstavljale iste lastnosti kot tiste, ki so v jedru. Te pogoje je bilo mogoče doseči z dvema oblikovanima diamantom, s katerima so stiskali vzorce v velikosti praha ter z laserjem segreli vzorce na zelo visoke temperature. Uporabne rezultate so začeli pridobivati le takrat, ko so začeli dodajati več kot dva različna materiala. Ti eksperimenti so večinoma vsebovali silicij in kisik ter dve različni zlitini, saj so znanstveniki trdno verjeli,

da sta silicij in kisik prisotna v jedru. Ko so preučevali novonastale spojine, so opazili, da sta se kisik in silicij združila in tvorila silicijev dioksid (SiO_2), kar je ista zgradba kot mineralni kremen, ki se ga najde na površju Zemlje. Med preučevanjem so tudi ugotovili, da bi kristalizacija silicijevega dioksida lahko delovala kot izjemen vir energije za Zemljino magnetno polje.

Ti rezultati so zelo pomembni, saj pomagajo pri razumevanju energetike in evolucije jedra. Znanstveniki so v raziskavi tudi preučili, kakšno je bilo stanje Zemlje, ko je nastajala. Kristalizacija bi spremenila zgradbo jedra, saj bi se kisik

in silicij združila v kristaliziran kremen »quartz«, a ta proces ne more iti v neskončnost, saj bo prej ali slej zmanjkalo kisika ali silicija. Kristali ne morejo nastajati brez kisika ali silicija, a to nam samo pomaga pri ugotovitvi koncentracije silicija in kisika v jedru, saj so le določena razmerja kompatibilna z eksperimenti.

Vira:

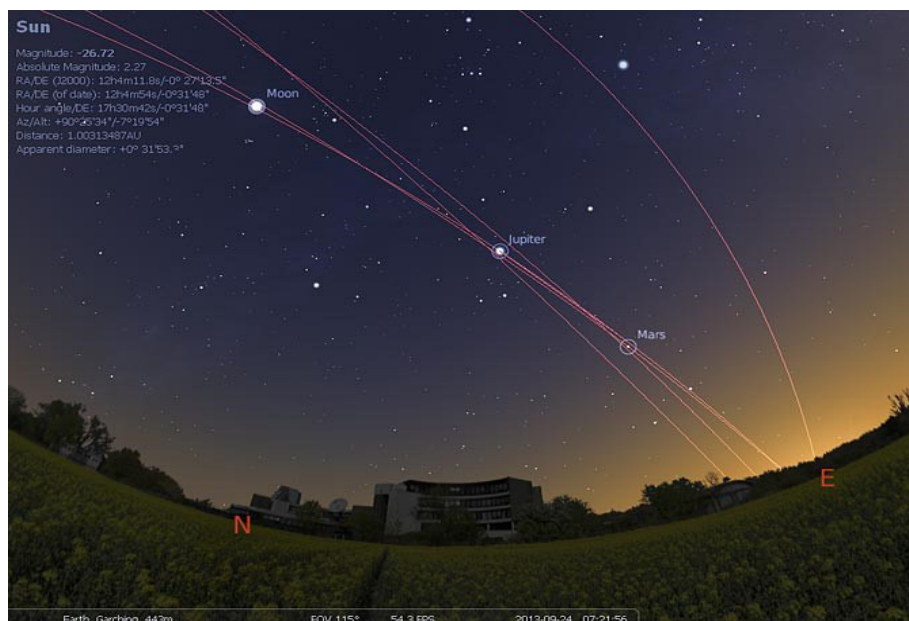
<http://www.titech.ac.jp/english/news/2017/037545.html>

<http://www.sciencetimes.com/articles/9341/20170223/quartz-crystals-at-the-core-of-earth-are-the-power-house-of-magnetic-field.htm>

Stellarium

Jure Sečnik, R 1. C

(Mentor: Aleš Volčini, prof.)



Zaslonski posnetek programa



Zaslonski posnetek stellariuma na mobilnem telefonu

Uvod

Stellarium je brezplačni GPL-namizni program, ki uprizaro realistično nebo in ozvezdja v resničnem času z OpenGL. S Stellariumom v resnici vidimo tisto, kar lahko opazujemo z očmi, daljnogledom ali s teleskopom. Stellarium uporabljamo tudi v planetarijih. Je preprost

za uporabo: vse, kar potrebuješ, so koordinate ali pa preprosto nastavimo čas in pogledamo, kakšno bo nebo v prihodnosti.

Stellarium lahko uporabljamo v različnih operacijskih sistemih, kot so npr. Linux, OS X, Windows 32-bit, Windows 64-bit in ubuntu. Stellarium nam je na voljo tudi na

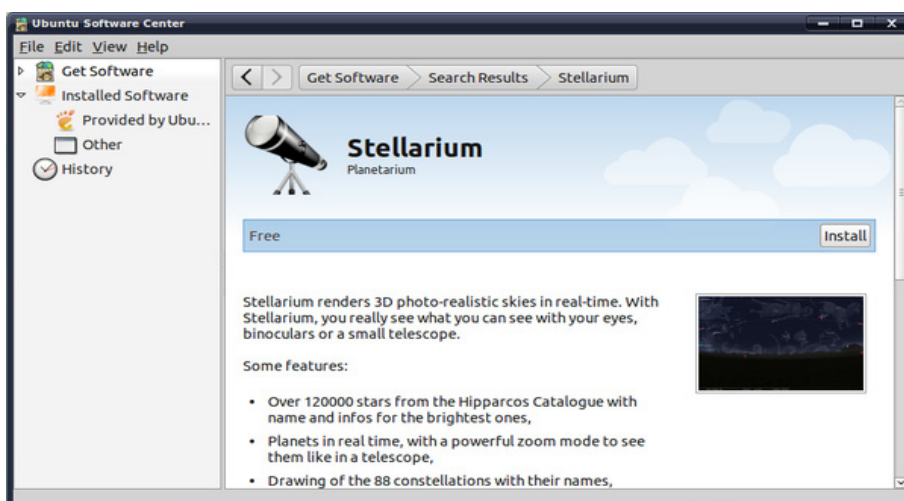
telefonu za androide in iOS. Edina njegova slabost je, da ga moramo kupiti za telefon.

Kratka zgodovina

Fabien Chéreau, francoski računalniški programer, je avtor stellariuma. Z njim se ubada večinoma v prostem času, razvija pa ga že od



Fabien Chéreau



Stellarium v Ubuntu software centru



poletja 2000. Prvi izvod stellariuma se je pojavil že leta 2001, stabilen izvod pa šele 31. julija 2016. Program je napisan v C++, namenjen pa je izobraževanju in učenju. Zaradi tega je program Stellarium (verzija 0.7.1) leta 2006 prejel zlato nagrado v kategoriji izobraževanja na tekmovanju Les Trophees du Libre free Software competition.

Predelana verzija stellariuma je bila uporabljena v projektu MeerKAT kot virtualna projekcija neba, ki je pokazala, kam so bile usmerjene antene radioteleskopov. Leta 2011 je bil stellarium dodan kot ena izmed bolj uporabljenih aplikacij v Ubuntu Software centru.

Sistemske zahteve

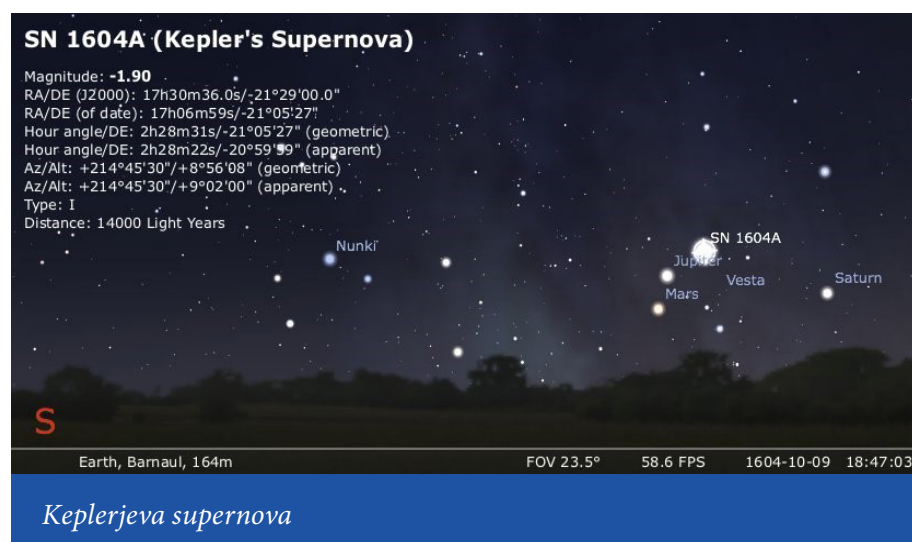
Minimalne zahteve

Program Stellarium za delovanje ne potrebuje veliko. Lahko imamo sistem Linux, Unix, Windows 7 in različice po njem, OS X 10.8.5 itd. V pomoč nam pride še 3D-grafična kartica, ki podpira OpenGL 3.0, potrebujemo pa tudi 512 MiB RAM-a in še 250 MiB na disku.

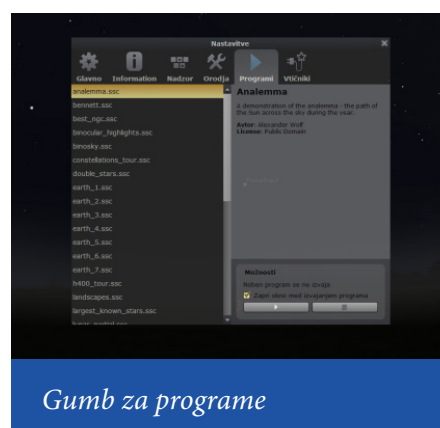
Priporočene zahteve

Priporočene zahteve so precej podobne minimalnim. Priporočeno je, da imamo sistem Linux, Unix, Windows 7 ali novejša različica, OS

X 10.8.5 ali novejša različica. Potrebujemo 3D-grafično kartico, ki podpira OpenGL 3.3. Priporočeno je tudi, da imamo na voljo 1 GiB RAM-a in 1.5 GiB na disku.



V obeh primerih je že dovolj, da imamo za normalno delovanje stellariuma povsem navadne standardne računalnike, kar je odlično, saj



Gumb za programe

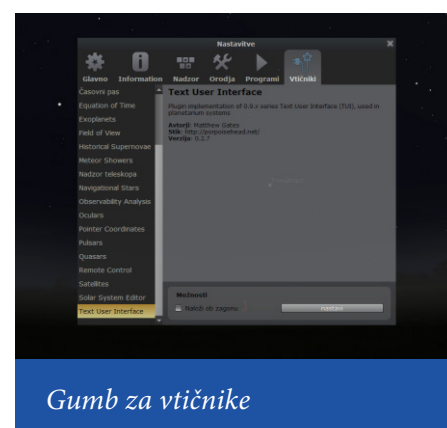
to pomeni, da lahko povsem vsak uporablja ta program, se iz njega tudi uči in vedno spozna kaj novega.

Namen

Stellarium je bil ustvarjen kot program za astronomijo, njegov namen pa je, da se naučimo o zvezdah, ozvezdijih in če ne moremo videti čisto nebo z zvezdami, ga lahko preprosto gledamo na računalniku. Pomaga nam tudi pri izobraževanju in učenju, zato je program dobil že omenjeno zlato nagrado za izobraževalno kategorijo.

Lastnosti in funkcije programa

Program Stellarium že na začetku vsebuje zvezdno mapo z več kot 600000 zvezdami. Lahko nadzoru-



Gumb za vtičnike

jemo čas, približamo pogled zvezd, vidimo planete in njihove naravne satelite, lahko pa je poudarek na estetiki programa, kot npr. konstelacije imajo svoje ilustracije, zvezde utripajo in vidimo tudi zvezdne utrinke. V programu so na voljo tudi razne simulacije, kot npr. simulacije supernov in Luninega oz. Sončevega mrka.

Program omogoča tudi sodelovanje uporabnikov tega programa in javnosti, ker lahko ustvarijo svoje vtičnike in manjše programe.

Program v svojem oknu za nastavitve omogoča, da si ogledamo vrsto programov, ki so v bistvu kratki videi oz. animacije. Te so ustvarili razvijalci Stellariuma. Večino teh programov pa v bistvu prihaja od

uporabnikov programa. Animacije so lahko karkoli: potovanje Sonca po nebu med letom, razne simulacije supernov, ozvezdja, pogled z navadnim daljnogledom itd. Programi imajo tudi svojo vrsto licenciranja, in sicer večina jih spada v javno last.

Poleg gumba za programe je še eden, ki nam omogoča ogled vtičnikov. Te so ustvarili razvijalci, večina pa jih je kar od uporabnikov programa kot že v prej omenjenih programih. Vtičniki imajo večji vpliv na estetiko programa kot pa na njegove funkcije. Vendar niso tako neuporabni, saj brez nekaterih vtičnikov razne programske animacije morda sploh ne delujejo ali pa so neuporabne. Vtičniki pa lahko tudi vplivajo na upravljanje programa, lahko nam omogočijo,

da ročno nadziramo vse funkcije oziroma nam omogočijo besedilni uporabniški vmesnik.

Kako dobiti stellariumovo kodo

Stellariumovo kodo lahko dobimo na več načinov, kar je odvisno od tega, kaj rabimo. Kodo programa lahko najdemo na internetu, in sicer na dveh straneh: SourceForge in Launchpad. Razvijalci so jo s pomočjo pregledne revizije shranili na splet v skladišče (Launchpad) in je prosto dostopna. Stran Launchpad lahko omogoči anonimnim uporabnikom, da preprosto prenesejo kodo, zato lahko sami odpravimo kakšne napake v programu in jih nato oddamo, da lahko razvijalci naprej razvijejo novo različico programa, ki bo brez teh napak.

The screenshot shows the Stellarium Launchpad web interface. At the top, there's a navigation bar with links like Overview, Code, Bugs, Blueprints, Translations, and Answers. Below this, a message states: "You can browse the source code for the development focus branch or get a copy of the branch using the command: `git branch lp:stellarium`". It also mentions "There are download files available for Stellarium." and "Stellarium has 5 active reviews." Below this, it says "Stellarium has 64 active branches owned by 23 people and 2 teams. There were 132 commits by 5 people in the last month."

The main section is titled "Bazaar branches" and includes a filter "Branches with status: Any active status" and a sort option "by most interesting". Below this is a table listing various branches:

Name	Status	Last Modified	Last Commit
lp:stellarium	Development	2 hours ago	8771. clean up of the code
lp:stellarium/trunk, 1.x			
lp:stellarium/0.15	Mature	2016-10-09	8590. Sync with trunk
lp:stellarium/0.14	Mature	2016-03-20	8058. Updated ChangeLog
lp:stellarium/0.12	Mature	2015-08-13	5619. Updated spectral classes of stars
lp:stellarium/stellarium/translations	Development	2016-10-25	6885. Launchpad automatic translations update.
lp:fschau/stellarium/opensslvidget	Experimental	2016-10-24	8480. Changed the way the OGL format is set
lp:stellarium/stellarium/SoCIS2016	Experimental	2016-10-16	8489. First implementation of the time-depe...
lp:sergekrivonos/stellarium/opensslvidget	Development	2016-10-16	8475. merge trunk
lp:stellarium/stellarium/toastimages	Development	2016-10-09	4867. sync with trunk (8726)
lp:gullame-chereau/stellarium/fix-orbits	Development	2016-09-05	8537. Attempt to support depth buffer on th...
lp:georg-zotti/stellarium/gz_planetAves	Development	2016-08-18	8138. Started work on improved planet aves...
lp:georg-zotti/stellarium/gz_AtmosphereTweaks	Experimental	2016-08-18	8118. Changed collar conversion from AdobeRG...
lp:nick-st2uz/stellarium/scope-uz	Development	2016-06-30	8489. Added Sync and Cancel buttons are add...
lp:cardino/stellarium/AddOn2	Development	2016-04-18	7556. Merge branch 'trunk'
lp:fschau/stellarium/scriptthead	Experimental	2016-04-12	8257. Implemented wait, waitFor and pausing...
lp:alewolf/stellarium/constellation-selection	Development	2016-04-07	8271. simplification of code
lp:petri-lubaneh/stellarium/stellarium	Development	2016-03-16	6943. working telescope control
lp:stellarium/stellarium/debian-packaging	Mature	2016-02-23	6674. Updated control file
lp:stellarium/stellarium/gz_reduceOpenGLStateChanges	Experimental	2016-02-04	8152. Stelainter: restore OpenGL state aft...
lp:stellarium/stellarium/OpenIP	Experimental	2016-01-26	8145. A first test of OpenIP usage (?)

Windows

- To get the full suite of external modules source code enter this URL in the Tortoise SVN check out from your development root directory as you did for the stellarium source. This will create a folder "stel-plugins" containing all the subdirectories of plugin modules sources.

```
https://stellarium.svn.sourceforge.net/svnroot/stellarium/trunk/extmodules @ stel-plugins
```

- Build the core Stellarium code according to the Windows Build Instructions page. Make sure the build sub-directory is called `build/msys`.
- Set the environment variable `STELROOT` to be the path of the stellarium source tree, for example:

```
export STELROOT=/c/msys/1.0/home/bob/stellarium
```

- Change to where you have the plugin source code installed, make a sub-directory `build/msys` and change into it.

```
cd /c/msys/1.0/home/bob/stel-plugins/<name of plugin>
mkdir -p build/msys
cd build/msys
```

- Run `cmake`, specifying the build type to be the same as that which was used to build the core code (Debug or Release). Then run `make`

```
cmake -G "MSYS Makefiles" -DCMAKE_BUILD_TYPE=Debug ../..
make
```

- Several of the plugins have a file called "installer.iss" in the main plugin directory. This can be used with the INNO installer generator to generate a windows installer for the plugin.

Tips

Creating the plug-in directory

If your plug-in stores its configuration data in a subdirectory of "modules" (in the user data directory), make sure that this directory exists. This can be easily done with the static function `StelFileMgr::makeSureDirExistsAndIsWritable()`.

For dynamic plug-ins, this is not strictly necessary, because to install the plug-in in the "modules" directory, such a sub-directory must be created by the install script. But when you are writing a static plug-in, or adapting a dynamic plug-in for static linking, don't forget that this time the directory does not exist by default!

Windows dynamic linking error

If a plugin uses a class that inherits a Stellarium class that contains the `Q_OBJECT` macro (for example, a dialog windows that inherits `StelDialog`), Stellarium will refuse to load it on Windows, returning an "Invalid access to memory location" error. The workaround is to copy the parent class (for example, `StelDialog`) to the plugin's sources and rename it appropriately.

Advice on how to solve this issue in another way will be much appreciated.

Podrobno opisan postopek ustvarjanja dinamičnih vtičnikov

Ustvarjanje vtičnikov

Že prej sem omenil, da ima stellarium možnost, da uporabniki sami razvijejo različne vtičnike za program. Ti so sestavljeni na dva načina, in sicer na dinamični in na statični način.

Dinamični vtičniki

Dinamični vtičniki so zelo preprosti. So v samostojnih dinamičnih arhivih, ki so naloženi takrat, ko stellarium deluje. To vtičnikom tudi omogoči, da jih delimo ločeno od stellariuma.

Statični vtičniki

Vtičniki so vgrajeni v stellariumove binarne datoteke. Te velikokrat uporabijo kot popravljene verzije nekaterih "uradnih" vtičnikov, skupaj z izdajo stellariumovih novejših različic. Obstaja tudi možnost, da ustvarimo samostojne statične vtičnike, ampak za to potrebujemo tudi po meri narejeno različico stellariuma. V tem primeru je veliko lažje, da naredimo dinamični vtičnik, ker je lahko samostojen. Če želiš, da nov vtičnik dodajo k

uradnim vtičnikom stellariuma, je najlažje, da ustvariš osebno vejo bazarja stellariuma v Launchpadu, ga razviješ in potem zaprosiš, da se osebna veja poveže z vejo glavne kode stellariuma.

Zaključek

Program Stellarium je preprost za uporabo, na voljo ima veliko funkcij, omogoča javnosti in uporabnikom, da dodajajo k programu, primeren je za učenje oz. raziskovanje in lahko se ga uporabi tudi v mednarodnih raziskavah ali projektih. Čeprav program živi od donacij oz. se preživlja z denarjem, ki ga dobi iz plačljive verzije na telefonu, mislim, da ima še kar dolgo in svetlo prihodnost svojega delovanja in uporabnosti. Sčasoma bodo programu dodali tudi nekatere nove funkcije, ki bi nam lahko pomagale pri razvoju astronomije in fizike.

Viri in literatura:

- <http://www.stellarium.org/> (27. 10. 2016).
- <http://bazaar.launchpad.net/~stellarium/stellarium/trunk/files> (27. 10. 2016).
- http://www.stellarium.org/wiki/index.php/Main_Page (26. 10. 2016).
- http://www.stellarium.org/wiki/index.php/How_to_get_Stellarium%27s_source_code (27. 10. 2016).
- http://www.stellarium.org/wiki/index.php/Plugin_Development (27. 10. 2016).
- http://www.stellarium.org/wiki/index.php/Stellarium_Wiki>About_Stellarium (27. 10. 2016).
- [https://en.wikipedia.org/wiki/Stellarium_\(software\)](https://en.wikipedia.org/wiki/Stellarium_(software)) (26. 10. 2016).
- <https://sourceforge.net/projects/stellarium/?source=directory> (26. 10. 2016).



Logotip programa

Zvezdno mesto – center ruske kozmonavtike

Neža Flisek, G 3. A

Zvezdno mesto je bilo zgrajeno leta 1960 v okviru ruskega vesoljskega programa. Že julija istega leta so se tam naselili prvi kozmonavti

vesoljske agencije, mora zadnji del priprav opraviti v Zvezdnem mestu, kjer se pred odhodom pokloni Juriju Gagarinu. Nato se odpravi



Slovenski srednješolci v Zvezdnem mestu blizu Moskve

z družinami. Mesto je bilo najstrožje varovano kot tajno vojaško območje. Glavni del je bil Center za pripravo kozmonavtov Jurija A. Gagarina. Poleg je bil še del, v katerem so bili delavci in kozmonavti z družinami; oba dela pa sta še vedno med seboj strogo ločena. Leta 1966 ga je v delno upravljanje dobila ruska vesoljska agencija Roscosmos, od oktobra 2008 pa ga ima v popolnem upravljanju. Prvi vodja centra za pripravo kozmonavtov je bil vojaški fizik in poročnik Jevgenij Karpov. Center za pripravo kozmonavtov ima infrastrukturo, opremo in naprave za usposabljanje kozmonavtov na vseh pomembnih področjih. Vsak kozmonavt, tudi kozmonavt tuje

na Bajkonur, od koder ga izstrelijo v vesolje.

Sovjetska zveza je v okviru vesoljskega programa prva dosegla veliko stvari, med drugim je bil sovjetski kozmonavt Jurij Aleksejevič Gagarin prvi človek, ki je obkrožil Zemljo. Jurij Gagarin se je rodil 9. marca 1934 in se izšolal za pilota vojaškega letala. Leta 1960 je bil



izbran v prvi odred kozmonavtov in 12. aprila 1961 kot prvi človek obkrožil Zemljo v vesoljskem plovilu Vostok 1; polet pa je trajal 108 minut. Umrl je 27. marca 1968 med treningom v letalski nesreči. Letalo MIG-15, s katerim se je ponesrečil Gagarin, je postavljeno pred Centrom za pripravo kozmonavtov kot opomin na tragično smrt heroja Sovjetske zveze in kot simbol povezanosti letalstva in kozmonavtike.

Vsi kozmonavti morajo pred poletom v vesolje opraviti šestletno usposabljanje, da se naučijo upravljati z Mednarodno vesoljsko postajo, preživeti v vesolju, postopke odhoda in vrnitve iz vesolja.



Praktične vaje na simulatorjih



Centrifuga CF-7

Priprave so razdeljene na tri dele: prvi del obsega splošne priprave, drugi del obsega tehnična znanja, v tretjem delu pa spoznajo vesoljsko plovilo, da se naučijo z njim upravljati in se seznanijo z Mednarodno vesoljsko postajo. Vsak del priprav poteka dve leti, v tem času pa morajo opraviti veliko medicinskih testov in testov znanja. Vsak kozmonavt, ki mu ruščina ni materini jezik, se mora naučiti ruščino, poleg tega pa mora znati odlično angleško. Kozmonavti največ treningov opravijo v vodnem bazenu, na delovnem modelu Mednarodne vesoljske postaje in v centrifugi. Tam opravljajo treninge večkrat v letu. Med treningi se vrtijo do ene minute, med izpitom od sedem do osem minut. Med vrtenjem morajo

ostati pri zavesti, saj morajo stalno pritiskati na gumb, prav tako pa jim med vrtenjem merijo velikost zenice. Treninge opravljajo v centrifugi CF-7, izpite pa v centrifugi CF-18. Maksimalna obremenitev med vrtenjem, ki jo lahko dosežejo, je 20 g, običajno pa dopustijo le 8 g, med izstrelitvijo pa obremenitev doseže 4,5 g. Treninge opravljajo tudi na simulatorjih, med katerimi je najpomembnejši simulator združitve Mednarodne vesoljske postaje in vesoljskega plovila Sojuz. Vaje opravljajo tudi v škatli z rokavicami, v kateri morajo z vesoljskimi rokavicami priviti vijak in računati na kalkulatorju. Po vrnitvi iz vesolja se začne rehabilitacija, ki poteka pol leta; v tem obdobju pa se ponovno naučijo pisati in hoditi, saj

se mišična masa med bivanjem v vesolju, kljub vsakodnevni telovadbi, ki traja 3 ure, zmanjša. Nekateri kozmonavti imajo srečo, da lahko v vesolje odpotujejo več kot enkrat, vendar morajo kljub že opravljenim pripravam, opraviti še dodatne priprave in teste.

Kozmonavti v vesolju preživijo do pol leta, med tem časom bivajo na Mednarodni vesoljski postaji, krajše MVP, v angleščini International Space Station, krajše ISS. MVP je skupni projekt šestih večjih vesoljskih agencij: ruske vesoljske agencije Roscosmos, severnoameriške vesoljske agencije NASA, evropske vesoljske agencije ESA, kanadske vesoljske agencije CSA, japonske vesoljske agencije JAXA in brazilske vesoljske agencije AEB. Predhodnik je bila ruska vesoljska postaja Mir, ki je v vesolju delovala od leta 1986 pa do sestrelitve leta 2001. Prvi modul MVP je bil v vesolje poslan leta 1998, prva človeška posadka pa je na postajo stopila leta 2000, ko je bil modulu Unity pridružen modul Zvezda, ki je omogočal človeško naselitev postaje, saj lahko posadka, v primeru odpovedi avtomatskega krmilnega sistema, celotno vesoljsko postajo upravlja iz modula Zvezda. Kasneje so bili pridruženi še ostali moduli, med njimi pa so najbolj znani moduli Pirs, Poisk, Cupola in ekspe-





Pogled iz modula Cupola

rimentalni modul Kibo vesoljske agencije JAXA. Mednarodna vesoljska postaja kroži okoli Zemlje na nizki Zemljini tirnici 360 km nad Zemljo, obkroži pa nas vsakih 92 minut. Na postaji je šest kozmonavtov iz različnih vesoljskih agencij, ki skupaj sestavljajo dve ekipi. Njihov dan je zapolnjen z delovnimi nalogami, ki jih morajo opraviti med osemurnim delovnikom. Vsak dan porabijo 3 ure za telovadbo, pol ure za umivanje, osem ur za spanje in ostanek časa za ostale dejavnosti. Ena izmed zanimivosti je ta, da morajo vodo pridobivati iz urina, saj je tovorna ladja ne pripelje zadosti za dva meseca, ko dobijo novo zalogo vode, hrane, kisika in svežih oblačil. Kisik morajo pridobivati iz vode, odvečen vodik pa spustijo v vesolje. Vsak kozmonavt potrebuje 4 litre vode na dan, od tega pol litra porabi za umivanje, en liter za pridobivanje kisika in dva litra in pol za prehranjevanje, saj mora hrano, ki je vakuumsko pakirana, zmešati z vodo, da postane užitna. Največjo nevarnost na postaji predstavlja kisik, saj se lahko že ob najmanjši iskri vse zaneti in eksplodira; prav tako pa morajo kozmonavti kisik, ki ga tovorna ladja pripelje v jeklenkah, porabiti do naslednjega prihoda tovarne ladje, v nasprotnem

primeru lahko prav tako eksplodira. CO₂, ki ga posadka izdihava, iz zraka izločajo s prečiščevalnimi sistemi, nato pa ga izpustijo v vesolje. Vodo morajo na postaji shranjevati za izredne razmere, na ameriškem delu jo konzervirajo z ioni joda, na ruskem delu pa z ioni srebra. Voda se tako lahko ohrani tudi 3 leta. Na postaji vlada breztežnost in kozmonavti se na posebne razmere navajajo tudi več tednov. Vsak kozmonavt po prihodu na Zemljo poudari, da je bila njihova najljubša dejavnost v prostem času opazovanje Zemlje skozi okna modula Cupola. Med bivanjem v vesolju kozmonavti opravijo tudi vesoljske sprehode, katerih namen pa je, nasprotno od naših predstav o opazovanju vesolja, delovne narave, saj morajo popraviti ali dodati dele na zunanjo stran vesoljske postaje.

Vsako vesoljsko plovilo, ki ga izstrelijo v vesolje, poganjajo raketni motorji, ki so nameščeni na dveh ravneh – prvi in drugi stopnji. Na prvi stopnji so štirje motorji, na drugi stopnji sta dva motorja, vsak od teh pa nosi 600 ton goriva, zato morajo biti zelo pazljivi pri izstrelitvi, da ne pride do eksplozije. Večino raketnih motorjev proizvede tovarna Energomash, v kateri že od

leta 1946 izdelujejo raketne motorje za ruski in tuji trg. Njeno drugo ime je tudi NPO Energomash V. P. Glushko, po inženirju Vladimirju Glushku, ki je bil glavni konstruktor raketnih motorjev do svoje smrti. V tovarni Energomash so najprej izdelovali raketne motorje za raketna letala in vojaške rakete, po vstopu SZ v vesoljsko tekmo pa so se usmerili v izdelavo motorjev za polet v vesolje. Danesasedajo 40 odstotkov svetovnega trga in 90 odstotkov ruskega trga, eden izmed njihovih glavnih kupcev je NASA. Njihovi najbolj slavni seriji motorjev sta seriji RD-170 in RD-180; motorji teh dveh serij so v vesolje ponesli obe vesoljski postaji in večino kozmonavtov, satelitov in vesoljskih teleskopov.

Ko se je človeštvo usmerilo v vesolje, se je začela zanj nova doba. Sedaj pa že razmišljamo o odhodu na Mars, rdeči planet, kar bo nov velik mejnik v življenju človeka.



Raketni motor

Od matematike ti lahko eksplodira glava!

Anej Svete, G 4. A

(Mentorica: Lidija Gorišek, prof.)

Pogosto slišimo o »nepredstavljivo velikih številih«. Novice in oglasi jih nenehno omenjajo, v šoli pa med številskimi množicami, vektorji, funkcijami in statistiko zanje ni časa. V primerjavi z ostalo šolsko snovjo so precej abstraktnjša, a vseeno zanimiva. V članku bom na kratko opisal to vejo matematike in odgovoril na očitno vprašanje: kako velika lahko števila res postanejo?

Najočitnejši odgovor je seveda neskončnost. Vendar si moramo, ko govorimo o največjih številih, najprej postaviti nekaj pravil, da ostanejo stvari pod nadzorom. Število mora biti uporabno oziroma mora nastati z določenim razlogom, zato ne velja največje število, povečano za 1. Prav tako ne velja neskončnost.

Zakaj? Neskončnost pravzaprav ni število, temveč le oznaka za nekaj zelo velikega. Okrog tega pojma kroži veliko paradoksov, ki pa nam lahko pomagajo razumeti nekaj računskih posebnosti. Če neskončnosti dodamo katero koli število ali celo še eno neskončnost, je seštevek neskončnost. Hkrati pa razlika dveh neskončnosti ni definirana, prav tako tudi ni množenje neskončnosti z 0 in deljenje neskončnosti z neskončnostjo.

Ker zdaj poznamo osnovna pravila, se lahko končno posvetimo pravim velikim številom. Prvo, eno najbolj znanih, je gugol z velikostjo 10^{100} . Zapišemo ga z 1, ki ji sledi 100 ničel. Po njem je poimenovano slavno podjetje

Google, ki se od angleškega zapisa tega števila (googol) razlikuje le zaradi pisarniške napake. A obstajata še večji sorodni števili, gugolpleks in gugolpleksion, ki predstavljata vrednosti 10^{gugol} in $10^{\text{gugolpleks}}$ oziroma 1, ki ji sledi gugol ali gugolpleks ničel. Zakaj pa so ta števila uporabna? Tu malo goljufamo, saj je njihov edini namen prikazovanje, kako velika so lahko števila. Največje količine v naravi se ne približajo niti najmanjšemu od teh treh števil. V človeškem telesu je namreč »le« 7×10^{27} atomov, na Zemlji jih je $1,33 \times 10^{50}$, v celotnem opazovanem vesolju pa 10^{80} .

Kako lahko potem sploh dosežemo gugol? Izkaže se, da ga lahko presežemo z igračo. Obstaja namreč zapletenejša verzija Rubikove kocke, ki jo lahko sestavimo na kar $1,95 \times 10^{160}$ možnih načinov. Mimogrede, navadna Rubikova kocka, tista s 26 manjšimi kockami, ima približno $4,3 \times 10^{19}$ možnih kombinacij.

Seveda pa obstajajo še večja uporabna števila. Precej znano je Grahamovo število. Je zgornja meja pri t.i. Ramseyjevi teoriji, zato namen gotovo ima, čeprav nam ga pravzaprav ni potrebno razumeti. To število je tako veliko, da ga ni mogoče predstaviti z navadnim matematičnim zapisom. Menda je celo tako veliko, da bi nam glava dobesečno eksplodirala, če bi si ga hoteli zapomniti, saj velja, da lahko v prostor določene prostornine shranimo le toliko informacij, kolikor jih lahko v črno luknjo enake

velikosti. Grahamovo število pa je preveliko celo za črno luknjo v velikosti naše glave. In čeprav je to število uporabno v višji matematiki, je njegov »oče« Ronald Graham bivši cirkusant.

Ko pa presežemo Grahamovo število, pridemo na področje nerazumljivo velikih števil, ki jih lahko opišemo le z besedami. Zanimivi sta predvsem dve: prvo je Loaderjevo število, imenovano po Ralphu Loaderju, ki je plod programa, ki ga je avtor napisal za tekmovanje v programiranju leta 2001. Zmagal je tisti, ki je v programskem jeziku C napisal program s 512 znaki ali manj, ki proizvede največje število, in zmagal je Loaderjev program. Drugo zanimivo število je Rayevo število. Tudi z njim je avtor, Agustín Rayo, zmagal na tekmovanju oziroma v dvoboju na univerzi MIT. Je z besedami podano število in od nastanka leta 2003 med matematiki velja za največje znano število. Definicija je sicer zapletena, toda tisti, ki v matematiki nismo najbolj vešč, jo lahko razumemo kot »najmanjše število, ki je še večje od prejšnjega«. Toda Rayo je to zapisal





(Mentorica: Lidija Gorišek, prof.)

Poznamo že vsa velika števila? Gotovo ne. Z današnjim razvojem znanosti in tehnologije bomo nedvomno prišli do še večjih uporabnih števil, ki si jih zdaj niti predstavljati ne moremo, in glede na dosedanje izkušnje do njih verjetno ne bodo prišli matematiki.

Predstavljaljaj si, da stojiš nekje na ekvatorju in imaš v rokah paket kart. Vsako sekundo jih premešaš v drugo zaporedje. Hodil boš okoli Zemlje, vendar z zelo počasnim tempom. Naredil boš namreč le en korak po ekvatorju vsako milijardo let. Ko narediš krog okoli Zemlje in se vrneš na mesto, na katerem si začel, odloži karte, vzemi kapalko in iz Pacifiškega oceana vzemi kapljico vode. Poberi karte in vajo ponovi. Vsako milijardo let naredi en korak in ko obhodiš Zemljo, ponovno vzemi kapljico iz Pacifika. To nadaljaj, dokler v Pacifiškem oceanu ni več vode. Sedaj vzemi navaden list papirja in ga položi na tla. In vajo spet ponovi. Ne pozabi, še zmeraj premešaš karte vsako sekundo v zaporedje, ki se prej še nikoli ni pojavilo. Nadaljuj s kroženjem okoli Zemlje, dokler ponovno ne izprazniš Pacifiškega oceana z eno kapljico, vsakič ko obkrožiš Zemljo. Ko ga spet izprazniš, položi še en list papirja na papir, ki je že na tleh, in nadaljuj s praznjenjem Pacifika, dokler ta stolp iz papirja ne doseže Sonca. Ko si ta stolp postavil, si zmešal karte v $3,85952 \cdot 10^{64}$ različnih zaporedij, kar pomeni, da še zmeraj nisi spremenil prvih treh cifer števila vseh možnih za-

Žal vam moram povedati, da ta primera v resnici nista izvedljiva in se ne bosta nikoli zgodila. Človek ne doseže starosti več milijard let, kakor tudi nima volje mešati kart več kot nekaj minut naenkrat. Ampak tudi če bi uspeli doseči nesmrtnost in imeli neskončno voljo, bi nam uresničitev prvega primera preprečilo Sonce, ki bi se Zemlji približalo na razdaljo, pri kateri bi izparela vsa voda, še preden bi naredil pet korakov.

Računalništvo je še boljše z matematiko

Veno Lan Banovšek, G 3. B

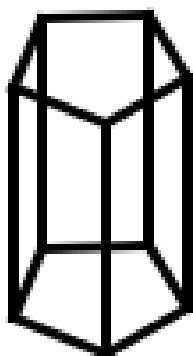
(Mentorica: Karin Kastelic, prof.)

Pri matematiki smo se v 3. letniku učili o geometrijskih telesih, med drugim o prizmah. Ker je programiranje moj hobi, je profesorica predlagala, naj napišem program, povezan z njimi.

Program sem si zamislil tako, da bi mu dodelil glavne potrebne lastnosti pravilne n-strane prizme, ki jo želim imeti. Program na podlagi tega izračuna ostale lastnosti, ki bi jih lahko potreboval, in izriše željeno prizmo v prostoru. Za program sem izbral javanski programski jezik, ki smo se ga učili pri predmetu računalništvo.

Najprej sem s pomočjo svojega znanja matematike poiskal osnovne lastnosti, ki jih potrebujem, da narišem prizmo in izračunam druge lastnosti: število kotov osnovne ploskve, dolžino njene stranice in višino prizme. Nadaljeval sem s programiranjem vnosa podatkov. Podatka, ki ju večinoma potrebujemo pri prizmah, pa sta površina prizme, torej seštevek plosčin vseh ploskev prizme, in prostornina prizme. Te program izračuna z uporabo enakih formul, kot smo jih uporabljali pri pouku matematike. Vnesel sem jih v program in naredil okno, ki izpiše vse podatke o prizmi. Pri tem se pokaže, kako zelo matematika olajša računanje pri programiranju, da ni treba računati »pešč«.

Toda program naj bi bil čim bolj zanimiv, zato sem ga izpopolnil, da prizmo tudi nariše v treh dimenzijah. S pomočjo matema-



Prizma: 5-kotnik

Stranica osnovne ploskve: 3,0 cm

Višina prizme: 10,0 cm

Površina prizme: $3096 + 150,0 = 3245 \text{ cm}^2$

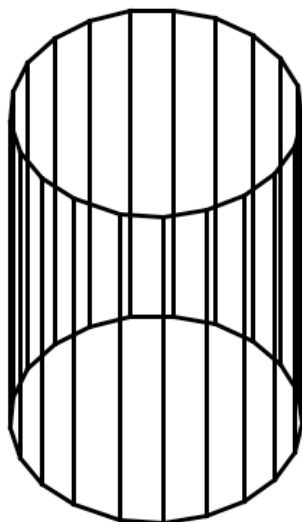
Prostornina prizme: 108390 cm^3

tičnih formul, ki smo se jih učili, sem iz podatkov o kotih in stranicah dobil točke v prostoru in iz točk prizmo, ki sem jo dal narisati programu. Dodal sem še zanimiv učinek, da se lik vrti, da ga lahko vidimo z vseh strani.

Toda pri matematiki smo se učili tudi o poševnih prizmah, ki so celo nagnjene. Za večji izziv sem se odločil sprogramirati program, ki bi risal in računal tudi s temi poševnimi prizmami. Tudi zanje sem uporabil matematično

znanje, ki smo ga pridobili pri pouku, in sprogramiral, da program nariše dano poševno prizmo, najde njeno os in jo okoli nje.

Tak program je primeren za skoraj vse prizme: poševne, pokončne, enakorobe, s katerokoli višino ali katerimkoli številom kotov (recimo 27-kotna osnovna ploskev). S pisanjem tega matematičnega programa sem se veliko naučil o programiranju v javi in poglobil razumevanje matematike.



Prizma: 21-kotnik

Stranica osnovne ploskve: 3,0 cm

Višina prizme: 30,0 cm

*Površina prizme:
 $63231 + 1890,0 = 65121 \text{ cm}^2$*

Prostornina prizme: 6702530 cm^3

Stožec je nezanimiv predmet – pa res?

Vildan Sejfuli, G 3. B

(Mentorica: Karin Kastelic, prof.)

Pri urah matematike se je naša profesorica odločila predati besedo nekaterim dijakom, da bi predstavili novo snov na zanimiv in izviren način. Tako sem se javil za predstavitev stožca.

Stožec je v matematiki (na kratko povedano) okroglo geometrijsko telo s krogom kot osnovno ploskvijo na izbrani ravnini in točko, ki leži kjerkoli zunaj te ravnine. Točke na krožnici tega kroga so nato z daljicami (ali krivuljami) povezane z zunanjo točko, ki se imenuje vrh. Lik, ki ga tvorijo te daljice (krivulje), se imenuje plašč. Najosnovnejši je pokončni stožec, katerega osni presek je poljuben trikotnik, stranska stranica je daljica, mreža plašča pa je krožni izsek.

Zanimivo je, da je lok plašča pokončnega stožca hkrati obseg osnovne ploskve, kar poenostavi računanje količin.

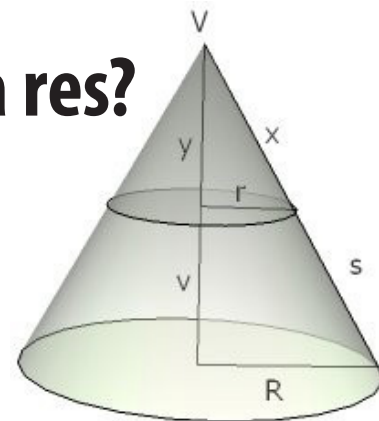
Formule za pokončni stožec so:

- | | |
|--|----------------------------|
| • $O = \pi r^2$ | ploščina osnovne ploskve |
| • $pl = \pi r s$ | ploščina plašča |
| • $P = O + pl = \pi r(r + s)$ | površina telesa |
| • $V = \frac{Ov}{3} = \frac{\pi r^2 v}{3}$ | volumen/prostornina telesa |

V poglavju o stožcu je del naše snovi še prisekani stožec. To je tisti, ki nima zgornjega dela. Sliši se nenavadno, ampak v resnici dobimo telo, katerega plašč povezuje dva različna kroga. To sta osnovni ploskvi telesa in zaradi tega se spremenijo tudi formule. Sošolcem sem predstavil izpeljavo za plašč prisekanega stožca.

Formule prisekanega stožca so:

- | | |
|---|---------------------------------|
| • $O_R = \pi R^2$ | ploščina večje osnovne ploskve |
| • $O_r = \pi r^2$ | ploščina manjše osnovne ploskve |
| • $pl = \pi s(R + r)$ | ploščina plašča |
| • $P = \pi(R^2 + r^2 + s(R + r))$ | površina telesa |
| • $V = \frac{\pi v}{3}(R^2 + r^2 + Rr)$ | volumen/prostornina telesa |



Izpeljava formule za plašč prisekanega stožca z uporabo slike zgoraj:

Plašču celega stožca odštejemo plašč zgornjega stožca:

$$pl = \pi R(s+x) - \pi r x = \pi(R(s+x) - rx) \quad (1)$$

Zaradi podobnih pravokotnih trikotnikov velja razmerje:

$$R:(s+x)=r:x.$$

Uredimo in dobimo:

$$x = sr/(R-r)$$

in vstavimo v (1):

$$pl = \pi \left(R \left(s + \frac{sr}{R-r} \right) - \frac{r^2 s}{R-r} \right) = \pi \frac{(R^2 s - r^2 s)}{R-r}$$

Ali ste vedeli, da:

- je valj (geometrijsko telo, katerega mreža plašča je pravokotni štirikotnik) v projekcijski geometriji v resnici stožec, katerega vrh je v neskončnost;
- obstaja v matematiki t. i. Gabrijelov rog ali Torricellijeva trobenta (Torricelli je bil znani italijanski matematik in fizik iz 17. stoletja), ki ima neskončno površino, ampak končno prostornino – od tod izhaja paradoks barve, saj lahko ta stožec z njo napolnimo, ne moremo pa ga pobarvati;
- da so mnogi naravni pojavi podobni stožcu: tornado, vulkani, morski vrtinci;
- beseda konus izhaja iz grške besede koniké, ki pomeni ošiljen oz. špičast;
- so volumen stožca lahko izračunali le z integrali, pri čemer so ponavadi uporabili ali stočničaste piramide ali neskončno krogov kot diske eden nad drugim?

Geometrija je obširen matematični pojem, najzanimivejša pa so ravno okrogla telesa. Stožec je eden izmed takih, in če bi se poglobili vanj, bi videli zelo veliko matematike za tako preprost predmet.

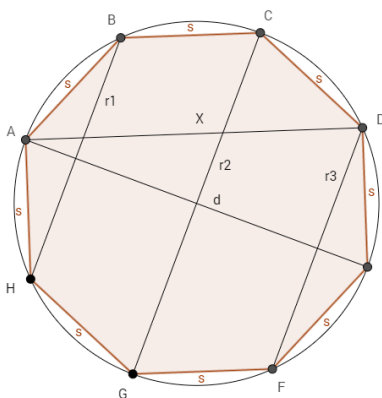
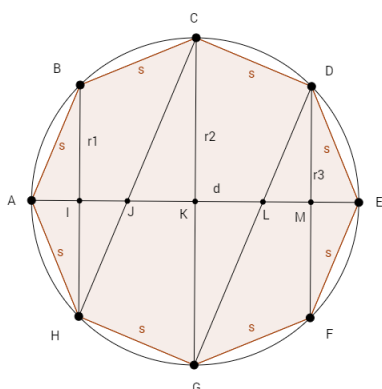
NESKONČEN KOTNIK ZA PI OKOLI OSI

David Kraševac, G 3. B
(Mentorica: Karin Kastelic, prof.)

Pri matematiki sem moral predstaviti kroglo. Že od nekdaj sem navdušen nad geometrijo, zato mi ni bilo odveč popestriti uro z izpeljavo formule za ploščino in volumen krogle. Način, kako sem se postopka lotil, pa je povezan z neskončnostjo oz. neskončnim kotnikom, tj. krogom.

Površina

Vzemimo polkrog in narišimo štiri enako dolge daljice, ki imajo oglišča na loku polkroga. Tako narišemo polovico osemkotnika. Vse skupaj zavrtimo za 2 Pi okoli premera d. Dobimo:



Naj bo jasno: če bi narisali neskončno takih daljic, bi dobili polkrožnice.

Če narišemo nekaj navpičnih daljic v dvodimenzionalnem prerezu krogle, dobimo nekaj podobnih pravokotnih trikotnikov.

$$\triangle ABI \sim \triangle JCK \sim \triangle LDM \sim \triangle ADE$$

Zaradi razmerij v podobnih trikotnikih veljajo enakosti:

$$r1/|AI| = r2/|JK| = r3/|LM| = x/s$$

Enačbe uredimo, obrnemo in seštejemo.

$$\begin{aligned} |AI| \cdot x &= r1 \cdot s \\ + |JK| \cdot x &= r2 \cdot s \\ + |LM| \cdot x &= r3 \cdot s \end{aligned}$$

Dobimo: (1)

$$(|AI| + |JK| + |LM|)x = s(r1 + r2 + r3)$$

Zaradi podobnih trikotnikov velja tudi: (2)

$$|AI| + |JK| + |LM| = r$$

Združimo enačbi (1) in (2) in dobimo:

$$(x \cdot r)/s = (r1 + r2 + r3) \quad (3)$$

Znova pogledajmo slike. Kot vidimo, smo dobili dva stožca in dva presekana stožca.

Zapišimo površino in upoštevajmo: (Pl - plašč)

$$Pl_{stor} = \pi r s$$

$$Pl_{(p. stor)} = \pi s(r_a + r_b)$$

Torej:

$$P = Pl_{stor1} + Pl_{(p. stor1)} + Pl_{(p. stor2)} + Pl_{stor2}$$

$$P = \pi s r1 + \pi s(r1 + r2) + \pi s(r2 + r3) + \pi s r3$$

$$P = \pi s(r1 + r1 + r2 + r2 + r3 + r3)$$

$$P = 2\pi s(r1 + r2 + r3) \quad (4)$$

Vstavimo enačbo (3) v enačbo (4):

$$P = 2\pi s((x \cdot r)/s)$$

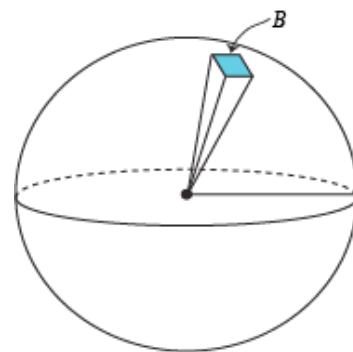
$$P = 2\pi x r$$

Naj gre število daljic n proti neskončnosti. Velja $x=2r$. Dobimo znano enačbo za površino krogle:

$$P = 4\pi r^2$$

Volumen

Za izpeljavo formule za volumen krogle bomo naprej kroglo razdelili na neskončno piramid.



Če upoštevamo, da gre število piramid v neskončnost, lahko enačimo polmer krogle z višino piramide. Zato za posamezne piramide velja: $V_p = 1/3 B \cdot r$. Zapišimo volumen krogle:

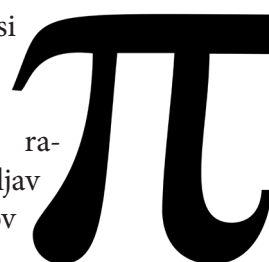
$$\begin{aligned} V_{krogle} &= V_{p1} + V_{p2} + \dots + V_{pn} = \\ &= 1/3 B_1 r + 1/3 B_2 r + \dots + 1/3 B_n r \end{aligned}$$

Vemo, da je seštevek vseh ploščin osnovnih ploskev piramid enak površini krogle. Tako dobimo:

$$V_{krogle} = 1/3 r (4\pi r^2) = 4/3 \pi r^3$$

In to je enačba za volumen krogle.

Moje delo pa ni povsem originalno. Ideje za postopke sem dobil na YouTube, in sicer na MathematicsOnline, kjer si lahko ogledate veliko zanimivih posnetkov razumljivih izpeljav formul, dokazov in pojasnil.



Ali je matematika lahko zanimiva?

Lidija Gorišek, prof.

Ko smo pred mnogimi leti brali v Vegovcu, kaj si učenci drugih šol mislijo o naših dijakih, so dijakinje Gimnazije Poljane izrazile spoštovanje do naših dijakov, rekoč: »Dobro znajo matematiko.« Za dijake, ki se zavedajo, da je matematika podporni predmet vsem strokovnim predmetom in zato

z zanimanjem dograjujejo svoje znanje, bi to potrdili tudi mi, profesorji.

Učitelji opažamo, da dijake zanimajo različne matematične teme, ki niso toliko povezane z učnim načrtom. Po elektronski pošti krožijo med njimi različni matematični

triki, nenavadni matematični zapisi in tudi zanimivosti.

Ker je med obravnavo predpisane učne snovi premalo časa za kakšne hudomušne matematične probleme oz. trike in za poglobljanje nekaterih znanj, lahko delček tega najdete za vsak letnik v izbranih temah v tem zapisu.

O ŠTEVILU 1089

Definicija:

Mnogokotniško število je v matematiki število, ki ga lahko razmesimo v obliki pravilnega mnogokotnika.

1089 je devetkotniško število.

Devetkotniško število dobimo z uporabo formule

$$\frac{n(7n-5)}{2}$$

za naravna števila n .

1, 9, 24, 46, 75, 111, 154, 204, 261, 325, 396, 474, 559, 651, 750, 856, 969, **1089**, 1216, 1350, 1491, 1639, 1794, ...

Matematik na oxfordski univerzi, David Acheson, je razkril nekaj zanimivosti o tem številu. Naslednji trije koraki vedno pripeljejo do števila 1089.

1) Vzemimo poljubno trimestno število, za katerega velja, da je razlika med prvo in zadnjo števk večja ali enaka 2. (npr. 237)

2) Zapišimo izbrano število v obratnem vrstnem redu števk (732) in od večjega števila odštejemo manjše. ($732 - 237 = 495$)

3) Dobljeni razliki prištejemo število, ki ima obratni vrstni red števk kot razlika, in dobimo 1089. ($495 + 594 = 1089$)

Za dokaz uporabimo nekaj preproste aritmetike:

Vsako trimestno število lahko zapišemo kot:

$$100 \cdot A + 10 \cdot B + 1 \cdot C,$$

število, ki ima števke v obratnem vrstnem redu, pa:

$$100 \cdot C + 10 \cdot B + 1 \cdot A,$$

za števke A ; B ; C velja $1 \leq A \leq 9$, $0 \leq B \leq 9$ in $1 \leq C \leq 9$.

Predpostavimo, da je $A > C$; (če $A < C$, najprej A in C zamenjamo).

Zgornji trimestni števili odštejemo:

$$(100 \cdot A + 10 \cdot B + 1 \cdot C) - (100 \cdot C + 10 \cdot B + 1 \cdot A) = 99 \cdot (A - C)$$

Razlika je vedno $99 \cdot (A - C)$.

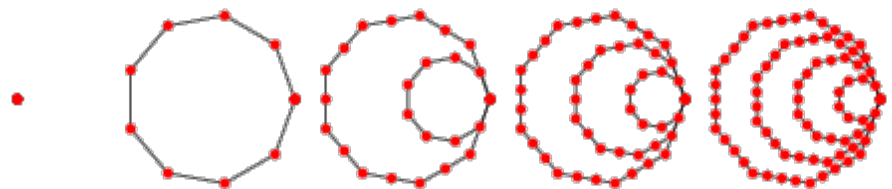
Zakaj mora biti razlika med prvo in zadnjo števk trimestnega števila večja ali enaka 2? Če bi izbrali tako trimestno število, pri katerem se prva in zadnja števka razlikujeta za 0 ali 1, bi bila razlika $99 \cdot (A - C)$ 0 ali 99 in število v naslednjem koraku ne bi bilo več trimestno število.

Nadaljujmo z dokazom 2. koraka, razliko $99 \cdot (A - C)$ bomo zapisali v drugačni obliki:

$$99 \cdot (A - C) = 99 \cdot [(A - C) - 1] + 99 =$$

$$(100 - 1) \cdot [(A - C) - 1] + 99 = 100 \cdot [(A - C) - 1] - 1 \cdot [(A - C) - 1] + 90 + 9 = 100 \cdot [(A - C) - 1] + 90 + 9 - (A - C) + 1 = 100 \cdot [(A - C) - 1] + 10 \cdot 9 + 1 \cdot [10 - (A - C)]$$

To je zapis trimestnega števila s prvo števk $(A - C) - 1$, druga je 9, tretja pa je $10 - (A - C)$.



Preostalo je dokaz 3. koraka:

Trimestnemu številu

$$100 \cdot [(A - C) - 1] + 10 \cdot 9 + 1 \cdot [10 - (A - C)]$$

zapišemo število s števki v obratnem vrstnem redu

$$100 \cdot [10 - (A - C)] + 10 \cdot 9 + 1 \cdot [(A - C) - 1]$$

in obe števili seštejemo.

Vsota je torej:

$$\begin{aligned} 101 \cdot [(A - C) - 1] + 20 \cdot 9 + 101 \cdot [10 - (A - C)] &= 101 \cdot [(A - C) - 1 + 10 - (A - C)] + 20 \cdot 9 = 101 \cdot [-1 + 10] + 180 = 1089. \end{aligned}$$

1089 se pogosto uporablja pri magičnih trikih, ker se ga lahko dobi iz dveh poljubnih trimestnih števil.

Ali najdeš kaj posebnega v stolpcih?

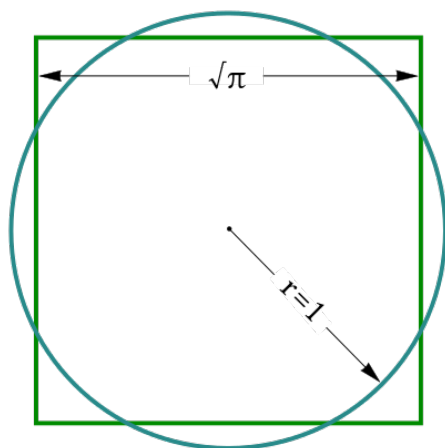
$$\begin{aligned} 1 \cdot 1089 &= 1089 \\ 2 \cdot 1089 &= 2178 \\ 3 \cdot 1089 &= 3267 \\ 4 \cdot 1089 &= 4356 \\ 5 \cdot 1089 &= 5445 \\ 6 \cdot 1089 &= 6534 \\ 7 \cdot 1089 &= 7623 \\ 8 \cdot 1089 &= 8712 \\ 9 \cdot 1089 &= 9801 \end{aligned}$$

KAJ JE KVADRATURA KROGA?

Kvadratura kroga

je znan problem klasične geometrije.

Kako konstruirati kvadrat, ki ima enako ploščino kot dani krog?



Naj bo polmer kroga r , njegova ploščina je $S = \pi r^2$. Kvadrat s str-

nico a ima ploščino $S = a^2$.

Iščemo stranico kvadrata a iz enačbe:

$$\begin{aligned} \pi r^2 &= a^2 \\ a &= \sqrt{\pi r^2} = r\sqrt{\pi} \end{aligned}$$

Bistvo naloge je natančno narisati stranico dolžine $\sqrt{\pi}$ z geometrijskim orodjem, tj. z uporabo šestila in ravnila.

Starogrški matematiki so dolgo neuspešno poskušali rešiti problem kvadrature kroga s šestilom in ravnilom, šele leta 1882 je bilo dokončno dokazano, da naloga z geometrijskim orodjem ni rešljiva. To je posledica dejstva, da je π *transcendentno število*, kar dokazuje Lindemann-Weierstrassov izrek. Transcendentno število je vsako kompleksno število, ki ni

algebrsko oziroma ni rešitev nobene polinomske enačbe oblike:

$$a_n x^n + a_{n-1} x^{n-1} + \dots + a_1 x^1 + a_0 x^0 = 0$$

kjer je $n > 0$ in so koeficienti a_i cela števila (ali racionalna števila), ne vsa enaka 0.

Kvadratura kroga kot prispevka

Kvadratura kroga je verjetno najbolj znana matematična naloga, ki se je s predpisanim geometrijskim orodjem ne da rešiti, zato je izraz »kvadratura kroga« postal prispevka za nerešljiv problem.

»Iskanje kvadrature kroga« pomeni jalo početje, ki je že vnaprej obsojeno na neuspeh, oz. je nekaj, kar je nemogoče.

CARDANOVA FORMULA

V tretjem letniku dijaki rešujejo tudi polinomske enačbe tretje stopnje. Med njimi so nekatere kar zahtevne in jih z elementarnimi metodami težko rešijo. Predstavila bom uporabo Cardanove formule za reševanje takih enačb.

Gerolamo Cardano (1501–1576) je bil rojen v Paviji, študiral je medicino in postal ugleden zdravnik. Napisal je več razprav o aritmetiki, astronomiji in fiziki. Ukvarjal se je z reševanjem enačb tretje stopnje.

Pred njim se je s takimi enačbami ukvarjal Niccolò Fontana (Tartaglia) (1499–1557).

Cardano je hotel na vsak način priti do njegove metode za reševanje kubnih enačb oblike:

$$\begin{aligned} x^3 + ax &= b \\ (a \text{ in } b \text{ sta celi števili}) \end{aligned}$$

Tartaglia je Cardanu razkril postopek reševanja enačb teh oblik pod pogojem, da ga ne bo razkril. Cardano je objavil najpomemb-

nejše delo *Ars Magna - Velika umetnost* (Artis magnae sive de regulis algebrae liber unus, 1545), v katerem je predstavil enačbe za reševanje korenov kubne enačbe, ki nosijo njegovo ime – Cardanove formule. Tartaglia se je zaradi tega zelo razburil in javno napadel Cardana.

Poleg Tartaglijeve oblike enačbe je Cardano rešil še enačbe oblike:

$$x^3 + b = a \cdot x, \quad x^3 + a \cdot x + b = 0, \quad x^3 = a \cdot x + b$$

Kako sta Tartaglia in Cardano reševala kubično enačbo?

1. korak:

Enačbo oblike $x^3 + ax^2 + c = 0$ prevedemo s substitucijo $x = y - \frac{a}{3}$ v enačbo $y^3 + py + q = 0$.

Postopek:

$$\left(y - \frac{a}{3}\right)^3 + a\left(y - \frac{a}{3}\right)^2 + b\left(y - \frac{a}{3}\right) + c = 0$$

$$\left(y^3 - 3y^2\left(\frac{a}{3}\right) + 3y\left(\frac{a}{3}\right)^2 - \left(\frac{a}{3}\right)^3\right) + a\left(y^2 - 2y\left(\frac{a}{3}\right) + \left(\frac{a}{3}\right)^2\right) + b\left(y - \frac{a}{3}\right) + c = 0$$

$$y^3 - ay^2 + \frac{a^2}{3}y - \frac{a^3}{27} + ay^2 - \frac{2a^2}{3}y + \frac{a^3}{9} + by - \frac{ab}{3} + c = 0$$

$$y^3 + \left(b - \frac{a^2}{3}\right)y + \left(\frac{2a^3}{27} - \frac{ab}{3} + c\right) = 0$$

Dobimo enačbo, ki nima kvadratnega člena $y^3 + py + q = 0$, kjer je

$$p = b - \frac{a^2}{3}, \quad q = \frac{2 \cdot a^3}{3^3} - \frac{a \cdot b}{3} + c$$

2. korak:

Rešitve enačbe $y^3 + py + q = 0$ iščemo v obliki $y = u + v$, kjer sta u in v novi neznanki.

Upoštevamo identiteto

$$(u + v)^3 - 3uv(u + v) = u^3 + v^3:$$

$$3uv = -p \text{ in } u^3 + v^3 = -q,$$

$$(u + v)^3 + p(u + v) + q = 0,$$



Gerolamo Cardano

$$u^3 + v^3 + 3uv(u + v) + p(u + v) + q = 0,$$

$$u^3 + v^3 + (3uv + p)(u + v) + q = 0,$$

$$(3uv + p)(u + v) + (u^3 + v^3 + q) = 0.$$

Postavimo zahtevo $3uv + p = 0$ in dobimo enačbo $u^3 + v^3 = -q$.

3. korak:

Za neznanki u in v velja: $u^3 \cdot v^3 = -p^3 / 27$, $u^3 + v^3 = -q$, z uporabo Vietovih formul pride-mo do kvadratne enačbe.

$$z^2 - (u^3 + v^3)z + u^3v^3 = 0 \text{ ali } z^2 + qz - p^3 / 27 = 0, \text{ kjer sta rešitvi}$$

$$z_1 = u^3 \text{ in } z_2 = v^3.$$

Kvadratno enačbo rešimo po znani formuli, kjer je $D = q^2 + 4(p^3/27)$ diskriminanta, in dobimo:

$$z_1 = u^3 = \frac{-q + \sqrt{q^2 + 4\frac{p^3}{27}}}{2} = -\frac{q}{2} + \sqrt{\frac{q^2}{4} + \frac{p^3}{3}}$$

$$z_2 = v^3 = \frac{-q - \sqrt{q^2 + 4\frac{p^3}{27}}}{2} = -\frac{q}{2} - \sqrt{\frac{q^2}{4} + \frac{p^3}{3}}$$

4. korak:

Upoštevamo, da je $y = u + v$. Izpeljali smo Cardanovo formulo:

$$y = \sqrt[3]{-\frac{q}{2} + \sqrt{\frac{q^2}{4} + \frac{p^3}{3}}} + \sqrt[3]{-\frac{q}{2} - \sqrt{\frac{q^2}{4} + \frac{p^3}{3}}}$$

$$y = \sqrt[3]{-\frac{q}{2} + \sqrt{\left(\frac{q}{2}\right)^2 + \left(\frac{p}{3}\right)^3}} + \sqrt[3]{-\frac{q}{2} - \sqrt{\left(\frac{q}{2}\right)^2 + \left(\frac{p}{3}\right)^3}}$$

Kljub temu da je Cardano v tistem času zelo pripomogel k reševanju enačb tretje stopnje, pa je sam priz-

nal, da včasih njihova uporaba ni tako enostavna. Poglejmo primer:

Rešimo enačbo: $y^3 + 3y - 14 = 0$.

Preberemo $p = 3$, $q = -14$, ju vs-tavimo v Cardanovo formulo in dobimo rešitev, ki po obliki ni tako preprosta, kot smo morda pričako-vali.

$$y = \sqrt[3]{7 + 5\sqrt{2}} + \sqrt[3]{7 - 5\sqrt{2}}$$

Da je vrednost te enačbe 2, na prvi pogled ni očitno. Zelo hitro enakost preverimo s kalkulatorjem ali po ne-kaj matematičnih zapisih. Naj velja:

$$\sqrt[3]{7 + 5\sqrt{2}} = a + b\sqrt{2}$$

$$\sqrt[3]{7 - 5\sqrt{2}} = a - b\sqrt{2}$$

$$\left(\sqrt[3]{7 + 5\sqrt{2}}\right)\left(\sqrt[3]{7 - 5\sqrt{2}}\right) =$$

$$(a + b\sqrt{2})(a - b\sqrt{2}) =$$

$$a^2 - 2b^2 =$$

$$\sqrt[3]{(7 + 5\sqrt{2})(7 - 5\sqrt{2})} =$$

$$\sqrt[3]{49 - 50} = \sqrt[3]{-1} = -1$$

Če bi se poglobili v reševanje Pello-ve enačbe $a^2 - 2b^2 = -1$, bi ugotovi-li, da je:

$$(1 + \sqrt{2})^3 = 7 + 5\sqrt{2} \text{ ali } \sqrt[3]{7 + 5\sqrt{2}} = 1 + \sqrt{2}$$

$$(1 - \sqrt{2})^3 = 7 - 5\sqrt{2} \text{ ali } \sqrt[3]{7 - 5\sqrt{2}} = 1 - \sqrt{2}$$

kar nas privede do enakosti

$$\sqrt[3]{7 + 5\sqrt{2}} + \sqrt[3]{7 - 5\sqrt{2}} = 1 + \sqrt{2} + 1 - \sqrt{2} = 2$$

Veliko težje pa je s pomočjo Car-danove formule iskati kompleksne rešitve. Svoje znanje bi morali poglobiti v vsebine polarne zapisa kompleksnega števila in v korene enote. Ker pa vemo, da z eno re-alno rešitvijo y , ki nam jo ponuja tudi Cardanova formula, znamo poenostaviti zapis enačbe $y^3 + py + q = (y - y_1)(y^2 + cy + d) = 0$ in rešiti kvadratni del po elementarni poti, lahko poglobljanje znanja po-čaka do študija na kakšni tehniški fakulteti.

Prvi Slovenec s Pulitzerjevo nagrado

Srdjan Živulović – nekdanji vegovec, ki pa ga je elektrika preveč tresla

Matevž Drnovšek in Tine Venko, G 2. B

Človek, poln zanimivih zgodb iz svojega življenja in potovanj po svetu. Begunci, vojne, osebni fotograf ... Dobitnik Pulitzerjeve nagrade, s katero se je vpisal v zgodovino kot prvi Slovenec. Začel pa je z Vegovo, a ga je preveč »tresla«, zato se je odločil za fotografsko smer.

Imeli smo priložnost in obenem čast, da smo se lahko pogovarjali z njim na Vegovi.

Minilo je dobro leto, odkar ste bili Vi in ekipa iz Reutersa nagrajeni s Pulitzerjevo nagrado. Kako ste se počutili, ko ste izvedeli, da ste prejemnik take prestižne nagrade?

Zvečer me je iz Berlina klical moj šef, malo pred deseto uro in dejal: »Žiga, dobil si pulicerja, čestitam!« Jaz pa njemu nazaj: »Ne mene danes hecat, sem imel že cel dan poln, ne me zdaj še ti!« Šef mi je rekel: »Ne ne, resno mislim, pojdi pogledat na internet pa boš videl!« Kar nisem verjel in sem še dobre pol ure ležal na kavču, dokler me niso premagale besede šefa in sem šel do računalnika, kjer sem res izvedel, da sem eden izmed prejemnikov. Ostalim družinskim članom sem rekel, naj pridejo v spalnico, da ne bom trikrat govoril ... »Prejel sem pulicerja, to je to!« Ob takem uspehu pa seveda ni šlo brez zdravice. Deset let smo se med kolegi hecali na tiskovnih konferencah, ko sem jim pokazal najslabšo fotografijo, da bo prav ta za pulicerja ... in evo deset let pa je res prišla!



Nagrajena fotografija - prvi dan migrantskega vala zjutraj v Rigoncah

Ali nam lahko poveste več o sliki? Kakšne so bile razmere? Je to vaša najboljša fotografija?

Pulitzerjeve nagrade na Univerzi Columbia v New Yorku podeljujejo vsako leto, in sicer v 14 novinarskih kategorijah in petih knjižnih in v eni glasbeni in dramski kategoriji. Leta 2016 sem bil med nagrajenci tudi jaz, skupaj z ekipo tiskovne agencije Reuters s serijo 17 fotografij, ki prikazujejo balkansko migrantsko pot. Fotografija, ki je bila posneta 20. oktobra 2016, predstavlja prvi dan migrantskega vala zjutraj v Rigoncah. Preden fotografiram, si zmeraj malo predstavljam v glavi, in ker se svet tehnologije iz dneva v dan spreminja, fotografijo posnamem, potem pa samo pritisnem na gumb in že odleti v uredništvo. Ob enih me je klical šef in mi dejal, da je fotografija objavljena v vseh medijih, in nadaljeval, da grem lahko domov, a tega nisem mogel, temveč

sem pomagal migrantom, saj tisti dan še ni bilo Rdečega križa, ampak samo policija. V bistvu si sploh ne znam predstavljati, ali je to moja najboljša fotografija, ampak iz te serije fotografij pa zagotovo je!

Ste priznani reportažni fotograf in edini slovenski fotograf s Pulitzerjevo nagrado. Kdaj ste začutili ljubezen do fotografiranja?

(Nasmeh) ... Za enkrat sem še edini, ja. Ljubezen do fotografije? Tamle čez obstaja ena trgovina, Maksimarket, in moj oče mi je za novo leto pri trinajstih letih kupil prvi fotoaparatus in od tistega momenta ni šel od mene. Starši so mi vsake toliko časa kupili kakšen film in potem sem ga peš nesel razvit h gospe Zofki (Foto Zofka), da mi je razvila in naredila kakšno fotografijo. Čez en teden pa spet peš po fotografije. Moj prvi fotoaparatus pa je bila berreta, ki jo hranim še zdaj.

Kdaj in kako ste spoznali, da lahko svoj hobi, fotografiranje, spremenite v kariero?

Takrat še nisem vedel, da obstaja fotografska šola in s kolegom sva se zmenila, da greva na elektro šolo – Vegovo. On je naredil sprejemne za šibki tok, jaz pa ne in sem pristal na jakem toku ... To mi pa ni šlo in čez pol leta sem obupal nad Vegovo. Kot otrok sem vsak torek na štirinajst dni kupoval Avto revijo in potem se je v meni porodila želja, da bi delal za Avto revijo, kar se mi je tudi uresničilo. Nato sem delal prakso pri časopisu Delo in tam so se vse moje sanje uresničile.

Kaj Vas pritegne na fotografijah? Motiv, zgodba, kompozicija ...

Pri meni? Ne znam si razložiti, meni je bistvo vsega, vse zgodbe, da je fotografija meni vseč. Ni važno, ali je umetniška ali ni, reportažna me ne zanima. Meni je bistvo samo to, da je meni vseč, in če je, jo pokažem naprej. Od tukaj naprej pa si lahko mislijo, da je zanič, slabo obrezana, narejena, meni ni več pomembno.

Kako Vaš poklic poklicnega fotografa vpliva na vsakodnevno življenje? Veliko potovanj? Imate fotoaparata vedno pri sebi?

Fotoaparat imam zmeraj pri sebi in, ja, tudi na dopustu ga imam zmeraj s seboj. Slikam lepe stvari, naravo, sončne zahode, živali ... To je moj način življenja. Če ne fotografiram, pa urejam slike, gledam, iščem, kaj drugi delajo. Pri nas doma smo malo »ubrisani«, vsi fotografiramo, razen srednji je prišel malo k pameti in je računalnikar. Seveda veliko slikam tudi svoje domače.

Ni veliko znanega o Vašem šolanju, zato Vas prosimo, da pojasnite naslednje vprašanje. Ugotovili smo, da ste bili dijak Vegove, a ne

ravno dolgo. Kje ste nadaljevali svojo pot?

Vmes sem se zadrževal v Kinoteki, kjer je bila ob desetih zjutraj matinea, in vse dobre filme sem videl tam. Starši so me imeli počasi zadosti in sem šel za pol leta delat v usnjarno v Šmartno pri Litiji, kjer sta bila moja dva zaposlena. Bližalo se je novo šolsko leto in poiskali so fotografski oddelek v Centru strokovnih šol za Bežigradom in mama je to odločitev podprla: »To je zate, delo z ljudmi in še aparat imaš, rad delaš to!« Odločitev se je obrestovala, saj sem prišel v stik s prakso na Delu in že prvi dan ob sedmih zjutraj sem si rekel: »To je to, jaz od tukaj ne grem več!« Po vojski sem se zaposlil na Zavodu za varilstvo kot tehnični fotograf in vsak vikend hodil na Delo, da bi dobil priložnost reportažne fotografije. Tam sem sedel in upal, da se me kdo usmili. Pri Delu sem bil stalno zaposlen od 1985 in ga vestno zapustil leta 1992. Zapustil sem Delo z napako, ki jo še vedno očitam takratnemu šefu Jocu Žnidaršiču. »Ti, poslušaj, Berlinski zid bo padu, gremo v Berlin.« Joco je odgovoril: »Ne ne ne, to je brez veze!« Od takrat Žiga več ne odlaša in če vidi priložnost, je ne izpusti.

Živulovičeva fotografska pot se je nadaljevala z ustanovitvijo lastne fotografske agencije in partnerstva s časopisom Republika. Delal je tudi za zunanje ministrstvo, predsednika vlade in predsednika republike kot uradni fotograf, in to celih 17 let. Tako opiše ta čas: »To je bilo tako lepo obdobje mojih ljudi, ker smo res videli ves svet, ampak res ves svet. Evropa je bila za dobro jutro, ni bilo privatnega aviona, da se mi ne bi vozili z njim.« Od leta 1991 dela tudi za agencijo Reuters, ampak pod zavezo, »da ne bom šel iz Slovenije za njih; če grem do Zagreba za kakšno traparijo, drugače pa še to ne.«



Srdjan Živulović (foto: Nejc Bertonec, G 2. A)

Imate mogoče kakšen nasvet za nadobudne vegovce, kako doseči svoje življenjske cilje?

Ena in edina zapoved je, da uživate v svojem delu, nič drugega. Ne glejte na denar, ne glejte na čas, ne glejte na to, da se ne splača, ampak samo gas. Delajte zase, vedno delaš samo in izključno zase. Ne zato, da boš šefu pokazal, kako je, ne glede ali boš ali ne boš zaslužil, to slej ko prej pride. Prideš do tega stanja, ali boš moral ali ne boš ali se splača, ampak delaj zase. Šolo delaš zase, zaposlitev delaš zase, moraš biti verodostojen sebi, najprej sebi, potem pa vsem ostalim, celemu svetu, in seveda delaj tisto, kar te veseli, tisto, samo na glavo, nič spraševati šefa, če lahko ali zakaj. Jaz bom to naredil!

Srdjan Živulović oziroma Žiga, kot se nama je predstavil, je človek mnogih zgodb, ki bi jih lahko poslušali cele dneve, in se jih ne bi naveličali. Prepotoval je ves svet, kar mu lahko marsikdo zavida. Pogovor z njim nama je pustil izjemen pečat, saj je poln življenjskih zgodb in modrosti.

Zahvaljujemo se Srdjanu Živuloviću za sodelovanje in zanimive zgodbe.

Grabljice, ki lahko pograbijo vse

Nina Dekleva, univ. dipl. bibl. in sin.

Kadar ljudje slišijo, da sem študirala kitajski jezik, običajno takoj sledi vprašanje: »A tiste njihove grabljice znaš tudi pisati?« Seveda! Za znanje jezika se je bilo potrebno naučiti tudi pisave, ki marsikoga spominjajo na nerazumljive čačke, ki nimajo smisla. A ravno nasprotno. V sodobnih pismenkah lahko prepoznamo marsikatero piktogram, iz katerih se je razvila kitajska pisava, zato kitajščino imenujemo tudi slikovni jezik. Najbrž bo vsak izmed vas iz slik prepoznal pomen naslednjih pismenk:



Pismenka 家 jiā (dom, družina) je sestavljena iz dveh delov: iz radikala (korena) 宀, ki ponazarja streho, in pismenke prašič 豕. S pujsom pod streho (živinorejo) se je ljudem oblikoval koncept doma. To, da smo že v prazgodovini začeli deliti svoj dom z živalmi, je prineslo možnost stalne naselitve.

Kitajščina je tonemski jezik, kar pomeni, da se besede, ki jih kitajščina pozna, berejo v različnih (natančneje štirih) tonih. Ena pismenka (zlog) je izvirno pomenila eno besedo, ki se je lahko izgovorila v štirih tonih (npr. zlog pī: 1. ton: pī, 2. ton: pí, 3. ton: pǐ, 4. ton: pì), glede na izgovorjeni ton pa je imela beseda različen pomen.

Preden se je kitajščina razvila v sodobno oz. mandarinsko kitajščino

(普通話), je bila enozložen jezik. Ker je kitajščina polna enakozvočnic, je bilo precej težavno razumevati govor in razlikovati homonime med seboj. Zelo nerodno bi bilo, če bi si naročili namesto piva pí (啤) arzenik pī (砒) in da bi se z nekom razšli pī (匹), namesto da bi se z njim ujeli pī (嫫). Težavnost klasične (enozložne) kitajščine dobro ponazarja zgodba o pesniku Shiju in kamnitih levih 施氏食獅史, ki je napisana v klasični kitajščini. Posebnost zgodbe je, da se vseh 92 pismenk izgovarja enako, tj. shi (izgovarja se podobno kot slovenski š), le da so besede v različnih tonih. Zgodbica je na papirju videti tako:

“石室詩士施氏，嗜獅，誓食十獅。施氏時時適市視獅。十時，適十獅適市。是時，適施氏適市。氏視是十獅，恃矢勢，使是十獅逝世。氏拾是十獅屍，適石室。石室濕，氏使侍拭石室。石室拭，氏始試食是十獅。食時，始識是十獅，實十石獅屍。試釋是事。” Transkripcija v latinico oz. pīnyīn (sistem zapisovanja kitajskih glasov oz. pismenk v latinici) pa bi bila videti (in slišati) tako:

“Shí Shì Shī Shì Shī Shì, Shì Shī, Shì Shí Shí Shī. Shì Shí Shí Shì Shì Shì Shī. Shí Shí, Shì Shí Shī Shì Shì. Shì Shí, Shì Shī Shì Shì Shì. Shì Shī Shí Shī Shī, Shì Shī Shì Shì, Shī Shì Shí Shī Shì Shì. Shì Shí Shí Shí Shī Shī, Shì Shí Shì. Shí Shì Shì, Shì Shī Shì Shí Shì. Shí Shí, Shī Shí Shì Shì Shī Shī. Shí Shí, Shī Shí Shì Shì Shī Shī. Shì Shì Shì Shí Shī, Shì Shī Shì Shì Shì Shì Shì Shì.”

Lahko si torej predstavljate, da je zgodba za poslušalca, ki pred seboj nima pismenk, precej nerazumljiva. Zato se je sodobna kitajščina spre-



Zgodovina pismenke, ki pomeni družino

nila iz enozložnega v dvožložni jezik. Kako pa nastajajo nove besede za termine, ki jih kitajščina prej ni poznala? Kako prevzemajo besede iz tujih jezikov? Način, kako besede prevajajo, bi lahko razvrstili v tri različne skupine: fonetični princip, pomenski princip in fonetično-pomenski princip.

Fonetičen princip

Pismenke transkribirajo glede na zven besed. V kitajščini poiščejo ustrezne zloge, ki zvenijo kot tujka, ki jo poskušajo zapisati, hkrati pa »zanemarijo« pomen, ki ga nosi pismenka, s katero zvok zapišejo. Če si kitajske zloge izgovorite naglas, boste najbrž ugotovili, da zvenijo precej podobno (običajno angleškim) tujkam.

谷歌 Gǔgē = Google

納諾 (纳诺) nànuò = nano-

安培 ānpéi = amper

吉咖 jíkā* = giga-(* se izgovarja kot ě)

黑客 hēikè = heker

Pomenski princip

Poiščejo že obstoječe pismenke v lastnem jeziku, s katerimi bi lahko opisali pojem ali zajeli smisel izvirne besede, in pismenke združijo, da do-

bijo pravi pomen. Tako lahko tvorijo zanimive kombinacije besed.

電腦 (电脑) diànnǎo: 電 *elektrika*, 腦 *možgani (elektronski možgani)* = računalnik

天線 (天线) tiānxiàn:

天 *nebo*, 線 *žica (žica, ki sega v nebo)* = antena

網絡公民 (网络公民) wǎngluò gōngmín: 網絡 *omrežje (angl. net)*, 公民 *državljan (angl. citizen)* = »Netizen«

筆記本 (笔记本) bǐjìběn: 筆記 *zapisovati (angl. note)*, 本 *vir, merilnik za knjigo (angl. classifier for books)* = notesnik (*angl. notebook*)

線圈 (线圈) xiànquān: 線 *žica*, 圈 *obkrožati* = tuljava

Fonetnično-pomensko

Ustvarijo kombinacijo fonetičnega prepisa tuje besede in pomenskega dela, pri čemer nekatere pismenke tvorijo zvočno podobo besede, ki zveni kot izvirnik, druge pa ujamejo pomen.

因特網 (因特网) yīntèwǎng: 因特 *yīntè (fonetik, ki zveni kot angleški inter)*, 網 (*pomenski del*) mreža = internet

存儲卡 (存储卡) cúnchǔkǎ: 存儲 (*pomenski del*) shranjevali, 卡 *kǎ (fonetik, ki zveni kot angleški card)* = spominska kartica

博主 bózhǔ: 博 *bó (fonetik, ki zveni kot angleški blog)*, 主 (*pomenski del*) gospodar = blogger

Kot vidimo, Kitajci nimajo nikakršnih težav pri prevzemanju in oblikovanju sodobnih znanstvenih in tehnoloških izrazov in lahko z grabljicami »pograbijo« vse.

Učil sem se. Sem se naučil?

Janja Žlebnik, univ. dipl. org.



V srednji šoli se dijaki srečujejo s predmeti, ki jih lahko v grobem razdelimo na družboslovne, naravoslovne in strokovne oz. tehniške. Pri družboslovnih predmetih se učijo iz besedil z branjem (na primer zgodovino), pri strokovnih rešujejo probleme oz. naloge (matematika, programiranje).

V enem od prvih letnikov računalništva sem dijake vprašala, kakšne težave imajo z učenjem in zakaj mislijo, da niso tako uspešni, kot bi lahko bili. Odgovore je oddalo 27 dijakov. Več kot polovica jih je navedla, da imajo težave z zbranostjo pri učenju. Nekaj jih je odgovorilo, da se jim v osnovni šoli ni bilo treba učiti, zato zdaj ne vedo, kako se tega lotiti. Zanimivo je, da je bilo zelo malo odgovorov, da jim manjka motivacije. Večina pa se uči malo sproti in nekaj več pred testi. Snov večkrat preberejo, nekateri si delajo povzetke, zelo redki miselne vzorce. Nekaj dijakov si pomaga z inštrukcijami že med šolskim letom. Domače naloge delajo manj, bolj uspešni pogosteje. Naši dijaki se učijo. Zagotovo. Koliko pa se naučijo? Kako znajo pokazati svoje znanje, ko jih preverjamo, sprašujemo, zahtevamo od njih izdelke?

Z učenjem dosežemo znanje, razumevanje česa. To pa prinese uspeh in s tem lažje življenje. Čas, ki ga porabimo za učenje, pa ni nujno merilo znanja. Ko dobijo dijaki v roke ocenjene preizkuse znanja, jih manj uspešni komentirajo nekako takole: »Bil sem preveč površen.« »Nisem se mogel spomniti.« »Zmanjkalo mi je časa.«

Prvi pogoj za vsako učenje so seveda urejeni zapiski. Kako pa se izogniti zgoraj opisanemu dogodkom in doseči boljše rezultate? S sistematičnim delom v štirih korakih, ki ga bom opisala v nadaljevanju. Metoda je primerna za matematiko, fiziko in strokovne predmete, ki zahtevajo miselno reševanje nalog.

1. Usmeritev

- Nalogo obvezno v celoti preberemo (nepopolno prebrano besedilo običajno vodi do napak in več poskusov rešitve).
- Ugotovimo, kateri podatki so najpomembnejši.
- S svojimi besedami ponovimo, kaj naloga zahteva (analiza in določanje ciljev).
- Skiciramo rešitev (analiza).
- Približno ocenimo rezultat (določanje ciljev).

2. Sistematično urejanje

- Pred začetkom reševanja si zamislimo potek dela, izberemo postopek (načrtovanje).
- Natančno se držimo načrta, da ne pride do neorganiziranosti.
- Pravilno računamo, pazimo,

da pravilno (pre)pišemo števila, merske enote, decimalke ...

- d.** Preverimo, ali smo upoštevali vse faze postopka (pregledamo korake reševanja in si zagotovimo možnost hitrega preverjanja rezultatov).

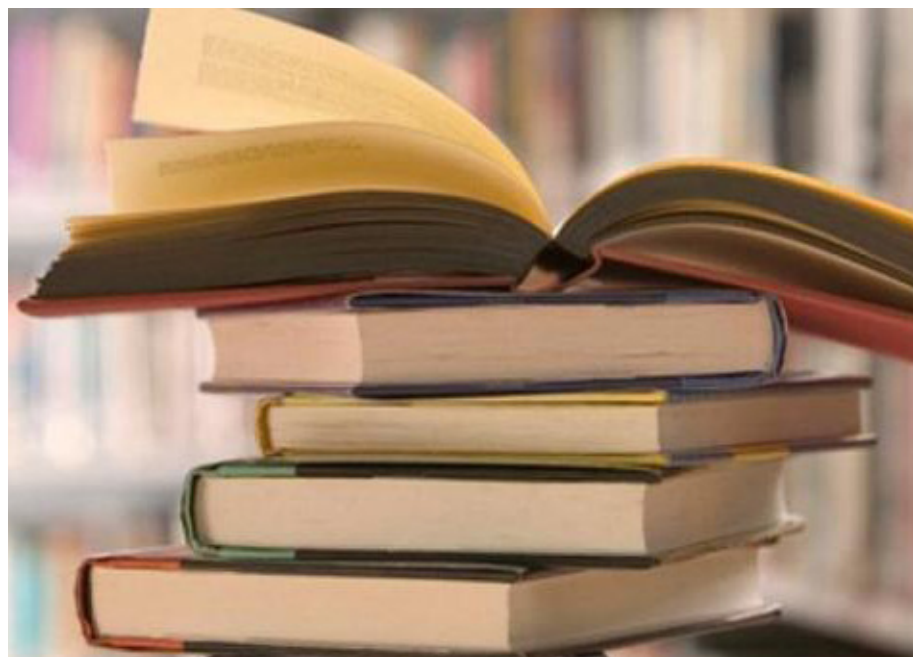
3. Vrednotenje rezultata

- a.** Spremljamo postopke.
- b.** Preverimo rezultat in ga primerjamo z oceno, ki smo jo postavili v prvem koraku.

4. Končni premislek

- a.** Napišemo rešitev v obliki odgovora na zastavljeno vprašanje.
- b.** Razmislimo o odgovoru glede na vprašanje v nalogi (ponovno preberemo vprašanje).
- c.** Premislimo, ali se rešitev uje s podobnimi nalogami, ki smo jih reševali med učenjem.

Učenje besedil oziroma pomnjenje pa zahteva nekoliko drugačen pristop, čeprav ga lahko podobno metodično razgradimo na posamezne korake. Branje namreč še ni učenje, je pa ključ do učenja.



Uporabimo naslednje korake:

1. Usmeritev

- a.** V mislih obnovimo predznanje.
- b.** Ponovimo pomembna dejstva (pomnjenje).
- c.** Pazljivo preberemo besedilo.

2. Načrtovanje

- a.** Ponovno pazljivo preberemo besedilo in skušamo razumeti vsebino.
- b.** Napišemo povzetek.
- c.** Postavljamo si vprašanja o vsebini.
- d.** Ugotovimo, če obstajajo nejasnosti, in poiščemo odgovore v drugih virih.
- e.** Odločimo se, ali bomo težje dele besedila prebrali ponovno.
- f.** Napišemo in razložimo neznane besede ali pojme.
- g.** Preberemo zapiske.

3. Izvedba

- a.** Na pamet ponovimo naučeno, uporabimo razlago vzrokov in posledic.
- b.** Vživimo se v vsebino, skušamo si jo predstavljati.



4. Spremljanje naučenega

- a.** Preverimo, ali naučeno razumemo ali smo si zapomnili vse pomembne informacije.
- b.** Poiščemo morebitne napake v znanju in jih odpravimo.
- c.** Ugotovimo, če obstajajo še nejasnosti, ali vse razumemo.

5. Izpopolnjevanje

- a.** Ponovimo na novo naučeno vsebino.
- b.** Ponovno preberemo le naslove in podnaslove besedila.

Tudi učenja se moramo naučiti, utrjevati metode in postopke, ki delujejo. Pomembno je, da ne preskočimo nobenega koraka. Postopki so preverjeni in lahko povečajo šolski uspeh do 40 %. Poskusite, spremljajte svoje učenje, ga ozavestite in se veselite svojih uspehov! Nikoli se ne nehajte učiti. Srečno!

Povzeto po: Joost Meijer, Marcel V. J. Veenman, Bernadette H. A. M. van Hout-Wolters: Metacognitive Activities in Text-Studying and Problem-Solving in osebnih zapiskih z delavnice Metakognicija, ki jo je vodil Marcel V. J. Veenman v Pragi 2015.

Matematični humor

Učitelj: Zapiši sinusni izrek. Kdaj ga uporabljamo?

Dijak: Ko ga rabimo.

Dijak: Profesorica, jaz sem vse znal. Zakaj sem dobil ena?

Učiteljica: Kaj si bil pa vprašan?

Dijak: Ja, tega pa ne vem.

Učitelj: Kaj so praštevila?

Dijak: Kakor že sama beseda pove, so praštevila števila, ki so jih uporabljali že v praveku.

Razumi, kdor more!

Vegovec definira pojme iz poglavja Funkcija in njene lastnosti:

Definicijsko območje:

So vsa realna št., ki so definirana.

So vsa števila, ki so definirana za to funkcijo.

Je realno št., ki mu pripada x.

Zaloga vrednosti:

je lastnost, ki velja za vse R.

Naraščanje funkcije:

Funkcija narašča, ko je rezultat izraču-

na $x_1 < x_2$ sledi $f(x) > x_2$ na koncu večji.

Funkcija narašča, ko je rezultat večji od največjega podatka.

So vsa realna števila, označimo jih z velikim M.

Medpredmetne povezave matematike z biologijo, s sociologijo, s psihologijo, z umetnostjo ...

Kotne funkcije so pediatrične.

pediátričen -čna -o prid. (á) nanašajoč se na pediatrijo: pediatrična klinika / pediatrična dežurna služba/

Seštevanje je asocialno.

asociálen -lna -o prid. (â-â) ki se noče ali ne more prilagoditi splošno veljavnim družbenim normam: bil je popolnoma asocialen človek / asocialni individualizem; zaradi asocialnih teženj ni našel zveze z okoljem, v katerem je živel; asocialno vedenje

Potenčna funkcija s sodim eksponatom je soda.

eksponát -a m (â) predmet za ogledovanje, postavljen na javen prostor, razstavni predmet: poslati eksponate za razstavo; muzejski, sejmski eksponat; sestaviti katalog stalnih eksponatov v galeriji/

NAGRADNA NALOGA: Tisti dijak/dijakinja, ki bo učiteljem matematike na šoli prvi povedal/-a, katere besede bi pravilno nadomestile podčrtane, dobi nagrado.

Sonetni venec

Ajda Erjavec Bartolj, univ. dipl. psih.

Dragi vegovci, drage vegovke,

z vami sem bila malo manj kot tri šolska leta in nekatere spremljala večino vašega časa na Vegovi.

Trenutni spomini so resni in igrivi, srce parajoči in šaljivi, grenki in sladki. Čez čas, upam, se bodo ohranili samo tisti, zaradi katerih mi je pri srcu toplo in bom delo na Vegovi zagotovo pogrešala. Verjamem, da se bomo še kdaj srečali ali celo sodelovali in obujali prijetne skupne spomine. In se tega veselim.

Dovolili ste mi, da sem vas dobro spoznala, po poti pa sem se veliko naučila tudi o sebi – za oboje sem vam hvaležna.

V vegovskem duhu šole ne morem zapustiti brez »izdelka«, zato sem svoje razmišljanje ob slovesu izrazila v rimah: kritičnih in optimističnih.

Vse dobro vam želim, dragi dijaki in kolegi, modro upravljajte s svojo dediščino in skrbno ravnajte drug z drugim.



I.

Alumnov stavba je veliko dala,
so tekom let številčno se nabrali:
pomembni, znani, mrtvi in ostali,
ki jih v mladosti avla je obdala.

Vprašanje, kaj je Vegova dodala,
to ni. Večinoma so svojo pot izbrali
in vsa neurja šolanja prestali –
vsaj té za večno je zaznamovala.

Pa tudi ône, ki so se zlomili
ali odšli, ker je prezapleteno
in je razum nemočen proti sili.

Za te ne sme nam biti kar vseeno,
sicer iz šole pamet smo spodili:
pravici mora biti zadoščeno.

II.

Pravici mora biti zadoščeno,
spregledana krivica je boleča.
Bolezen to je: past vsem nam preteča.
Bojim sveta se, kjer je vsem vseeno,

kjer ignoranca tvori sivo mreno,
odnosi pa kot neža so bodeča;
žal empatija misel je grozeča,
zato je lažje hraniti kopreno ...

... pasivno gledati zlorabo sile.
Kljub temu nam ne bo prizanešeno,
ko čas bo žetve takšne setve gnile.

Ponovna rast zahteva svojo ceno –
da človek lóti lastne se prisile.
A brez resnice ne bo odpuščeno.

III.

A brez resnice ne bo odpuščeno,
ker žrtve nujno rabijo resnico ...
... in ker je ta osnova za pravico.
Ta jemlje moč po krivem pridobljeno,

ki rasla z vsako bitko je dobljeno
tako kot dvom v resnico in pravico.
Ovita šola je v bodečo žico:
še veste ne, kaj vse je ogroženo.

Težko uméti je trenutno stanje ...
... je apatija dolgo gostovala.
Naj končno se zaključi zimsko spanje!

Resnica in pravica, to ni šala –
vrednote pretopiti v delovanje:
tako lahko ta šola bo obstala.

IV.

Tako lahko ta šola bo obstala,
če prostor bo za etiko, moralo.
Če argumentom se bo težo dalo,
res ustanova znanja bo postala.

Prosveta naj bi znanje negovala ...
A prav vsak znalec rabi ogledalo –
sicer postane vzvišeno budalo.
Po kom naj bi se mladež zgledovala?

Na tem je svetu znanja res ogromno,
ga glava ena ne bi obdržala ...
... in več, ko veš, bolj sklanjaš glavo skromno.

Če zbornica bi pristno spoštovala,
se šola bi razvijala prelomno
in svoje temelje še okrepčala.

V.

In svoje temelje še okrepčala
bo šola ta, če vas bo volja taka.
Zares uspešna bo pobuda vsaka,
če bo večina za njo glasno stala.

In če manjšina, ki je zdaj kričala,
sestopi s svoj'ga puhlega oblaka,
ki kvari vreme vsem, morí dijaka.
Bo šola višje standarde zadala?

Res upam, da zares tako bo –
da prav nikomur ne bo več vseeno,
če v vzgoji kdaj začuti pravo zlobo,

ki lomi dušo za korist nobeno.
Bo šola zaslužila to podobo,
če bo odprto dihala kot eno.

VI.

Če bo odprto dihala kot eno:
nebo je meja. Ne samo učenje,
je šola skupnost, vaja za življenje –
za to pomembno delo je pošteno.

Učitelj sam prodaja se pod ceno:
pasivno toži, pri spremembi len je ...
Pred drugimi prikriva svoje mnenje –
prastrah, kaj spet mu bode naloženo.

Kakovost dela je vprašanje sporno,
v pogovoru ste si zgradili steno.
Ta mora pasti. To bo še naporno.

Daj – vprašaj, reci in zamajaj sceno,
izzovi s svojo mislijo prodorno,
ne bodi sredstvo zgolj poleno.

VII.

Ne bodi sredstvo zgolj poleno;
obstaja tudi zdravo tekmovanje –
ironično morda – sodelovanje.
Vsak svoj je unikum in vsi smo eno.

Naj Janezek uči se – naučeno
bo Janez jutri spravil v delovanje.
Če Janezek zlorabljen in bolan je,
pripravljen za prihodnost ni nobeno.

In to je resna škoda, dragi moji.
Kakó bodočnost je poškodovala?!
Ubijajo in vzgajajo nas stroji,

namesto da bi skupnost vključevala.
Res smisel niso prazne bitke, boji,
ampak razum iz skupnega portala.

VIII.

Ampak razum iz skupnega portala
je tisti, ki spodbuja skupne sanje –
oplaja vednost, znanje in spoznanje –
zato je gred, ki naj razvoj bi gnala.

Na Vegovi je zbornica prodala
(za ljubi mir) ponos in prepoznanje,

da dostojanstvo ni finančno stanje,
da človek je zgolj bitje, ne pa skala.

Da moč človeška je z vezmi zgrajena
in nitka sama ne spreminja tkanja,
če je zadeva tesno prepletena.

Spremembe tkanja ni brez razdejanja –
potem tkanina znova je spletena.

Verjetno to ne gre brez uma, znanja.

IX.

Verjetno to ne gre brez uma, znanja,
da pamet vzgajali bi in gojili.
Smo cilje skupaj v družbi določili –
naj bi vodili do sodelovanja.

Naj bi predstavljali pravila panja
in smernice, kako naj bi učili,
da družbe zdrav razvoj bi spodbudili.
Za to pomembna so različna znanja.

Potrebno vplesti v vzorec je te niti,
da bi zares vzpostavili razumnost.
Kdor znanja nima, noče ... mora iti.

Prostora niti ni za ljubosumnost
med stroko, če ta hoče boljša biti ...
... enkratna res človeška je neumnost.

X.

Enkratna res človeška je neumnost:
sloni na čustvih, šibkosti razuma,
na strahu in odsotnosti poguma ...
... ni zgleda, le povod za ljubosumnost.

Učitelj sam zavira bistrornost,
če noče videti zakonov uma:
kolo so čustva in razum le guma –
konstrukcija je ključna za pogumnost.

Če glava prazna je, so prazne hlače!
Odveč so lažna hvala, praznovanja,
nihče pripravil prave ni jedače,

sadovi niso plod sodelovanja.
Potemkin za kulise skril je kače,
gradi se ne na lažni ceni stanja.

XI.

Gradi se ne na lažni ceni stanja;
promocija kot smoter je izdaja –
pozabljam na tisto, kar nastaja ...
... reklama ni cilj izobraževanja.

Mladina z apatijo zgolj odklanja,
kar jim odrasla družba zdaj prodaja ...
... in ta sivina brez topline traja –
v ušesih hrup iz medijev odzvanja.

So bolj kot kadarkoli prazni, sami,
jim manjka znanja, pada jim odpornost.
Odnos do sebe, drugih in med vami

vnebovpijoče kliče na pozornost.
Je čas za konec tej odvečni drami,
ocena prava kliče odgovornost.

XII.

Ocena prava kliče odgovornost:
če nek učitelj krut je – maščevalno
se znaša nad nekom, skupino – stalno,
edino prav je, da poplača spornost.

Je pedagoško dati zadnjo možnost –
če zadnja ni, to kmalu je banalno.
In je posmeh sistemu – »genialno« -
za bolne misli raj, za zlo priložnost.

Je plemenita naloga učiti ...
Ker je velika moč poškodovanja,
poslanstva tega ni za prepustiti

ljudem vprašljivega značaja, stanja –
pri tem jim treba je pomoč nuditi:
verjeli naj bi v moč sodelovanja.

XIII.

Verjeli naj bi v moč sodelovanja:
med vsemi, ne zgolj moškimi/ženami.
Če um odtehta tisto med nogami,
je šola prava ustanova znanja.

Menda, da nočemo možganov pranja –
vsak dan se širi floskula med nami.
Čemu v tla glavo rinemo in rami,
je v ogledalu prava slika stanja.

Deklet tu ni, ker te, ki so, trpijo ...
... pogosto objektivno nepravičnost.
Kljub temu nekatere preživijo,

ustvarjajo drugačno si resničnost –
k enakopravni skupnosti stremijo.
A tej resnica temelj je ... pravičnost.

XIV.

A tej resnica temelj je ... pravičnost ...
... če v sporih stroka, etika ni važna,
promocija zgolj maska manj sovražna,
a šola je sistemska brezobličnost.

In cilj postane mlačen vsa odličnost,
ki ceni jo avtoriteta lažna.
Res vzgoja ni produkcija metražna,
temveč poklic je plemenit – umetnost.

Me Vegova prav res je spremenila:
krivica v meni luknjo je izžgala,
ki jo podpora je in bo celila.

Tako sem skoraj vegovka postala
res srčno z vami cilje sem delila –
alumnov stavba je veliko dala.

Alumnov stavba je veliko dala,
pravici mora biti zadoščeno.
A brez resnice ne bo odpuščeno!
Tako lahko ta šola bo obstala

in svoje temelje še okrepčala,
če bo odprto dihala kot eno ...
Ne bodi glavno sredstvo zgolj poleno,
ampak razum iz skupnega portala!

Verjetno to ne gre brez uma, znanja.
Enkratna res človeška je neumnost –
gradí se ne na lažni ceni stanja!

Ocena prava kliče odgovornost:
verjeli naj bi v moč sodelovanja.
A tej resnica temelj je ... pravičnost.

Opravičilo z akrostihom

Staš Ščetinec, G 2. A

Al treba je blo oddat?
Lep članek, da do petka.
Enkrat, sredo, na busu zatežila mi je tetka ...
Natanko ko tuhtal sem, o čem pisat.

Končno, četrtek, skoraj spravu bi se delat.
In tako se izognil kazni metka.
Zdaj zdaj odklenjena delavnosti bo kletka!
Osel naivni, drznil sem si upat!

Rok oddaje mi že zdavnaj je odletel.
Kakor grešnik Raskolnikov v postelji ležim.
Oh, zdaj globoko resnico sem dojel ...

od usode tako nič ne dobim,
vseeno, če karkoli z življenjem bi počel
in če dovolj delaven sem, da v postelji ne obležim.

Vroče je

Ana Pohlin, R 3. A

Tu na Djerbi sonce sije, vroče je.
Kljub temu da veter vije, vroče je.
Cel ljubi dan na plaži poležavam,
zdaj pa že pozna ura bije, vroče je.
Vsake toliko se črni oblaki pokažejo
in kot iz škafa lije, vroče je.
Nikoli ne traja dolgo, se že sonce kaže.
Če je prevroče, te voda obvije, vroče je.
Niti se ni treba tunkat,
ker te sigurno kakšen val zalije, vroče je.
Potem mine teden in greš domov.
To te pa čist zbije, vroče je.



VEGOVA LJUBLJANA